



Universidad  
Carlos III de Madrid  
[www.uc3m.es](http://www.uc3m.es)

Departamento de Teoría de la Señal y Comunicaciones

Grado de Ingeniería de Sistemas de Comunicaciones

Trabajo de Fin de Grado

**Implementación de un receptor simple GSM**  
**mediante plataforma Software Defined**  
**Radio**

Autor: Isaac Morcillo Gutierro  
Tutor: Dr. Víctor Pedro Gil Jiménez  
Cotutor: Borja Genovés Guzmán

Leganés, Septiembre 2014

# **Agradecimientos**

A mis padres y mi hermano por su apoyo constante en casa.

A mis abuelas.

A mis compañeros de clase, especialmente a Adri y Carlos.

A mis compañeros de beca Sebas y Marta por motivarme a ser mejor ingeniero.

Y a mis tutores Víctor y Borja por la ayuda prestada.

# Resumen

Este trabajo consiste en la implementación de un sistema receptor-transmisor básico, basado en las estaciones base del estándar GSM. La implementación se desarrolla sobre una plataforma *Software Defined Radio* (SDR) haciendo uso de los transceptores de *National Instruments* USRP-2920 y del software LabVIEW.

Se presenta un receptor capaz de recibir la señal emitida por una estación móvil, de manera constante y estable, y de detectar ciertas secuencias de bits tras demodular los datos recibidos. La detección de estas secuencias de bits activa la transmisión de tramas GSM desde el transceptor USRP, en respuesta a la estación móvil.

# Índice

Índice de figuras

Índice de tablas

Glosario

## 1. Introducción

1.1. Experiencias previas

1.2. Objetivos

1.3. Contenido de la memoria

## 2. Planteamiento del problema

2.1. Estado del arte

2.2. Marco regulador

## 3. Cronograma

## 4. Estándar GSM

4.1. Visión general

4.2. Especificaciones

4.2.1. Bandas de frecuencia

4.2.2. Modulación

4.3. Arquitectura de la red

4.3.1. *Mobile Station*

4.3.2. *Base Station Subsystem*

4.3.3. *Network Switching Subsystem*

4.3.4. *Operation Support Subsystem*

4.4. Capa física

4.4.1. Tecnologías de acceso

4.4.2. Tipos de ráfagas

4.4.2.1. *Normal Burst*

4.4.2.2. *Frequency Correction Burst*

4.4.2.3. *Synchronization Burst*

4.4.2.4. *Access Burst*

4.4.2.5. *Dummy Burst*

4.4.3. Canales lógicos

4.4.3.1. Canales de tráfico

4.4.3.2. Canales de control

4.4.4. Estructura de las tramas

## 5. Entorno de trabajo

### 5.1. Hardware

5.1.1. NI USRP-2920

5.1.1.1. Receptor

5.1.1.2. Transmisor

5.1.2. Agilent ESG E4438C *Vector Signal Generator*

5.1.3. Agilent VSA 89600S

5.1.4. PC portátil

### 5.2. Software

5.2.1. LabVIEW

5.2.1.1. *LabVIEW Modulation Toolkit*

5.2.1.2. *LabVIEW MathScript RT Module*

5.2.2. Software VSA

## 6. Diseño

6.1. Interfaz de usuario

6.2. Sincronización del sistema

6.2.1. Sincronización en tiempo

6.2.2. Ajuste de frecuencia

6.3. Control del buffer de recepción

- 6.4. Mantenimiento de la sincronización
  - 6.4.1. Sincronización tras un número determinado de tramas
  - 6.4.2. Sincronización tras realizar una nueva recepción de datos
- 6.5. Procesamiento de datos
  - 6.5.1. Eliminación guarda
  - 6.5.2. Demodulación
  - 6.5.3. Detección del patrón
- 6.6. Transmisor
- 7. Pruebas y resultados
  - 7.1. Sincronización en tiempo
    - 7.1.1. Recepción de dos slots enviados por el generador de señales
    - 7.1.2. Recepción de ocho slots enviados por el generador de señales
    - 7.1.3. Recepción de tres slots enviados por un USRP
    - 7.1.4. Recepción de cuatro slots enviados por un USRP
    - 7.1.5. Recepción tras una nueva adquisición de datos
  - 7.2. Ajuste de frecuencia
  - 7.3. Eliminación del periodo de guarda
  - 7.4. Demodulación
    - 7.4.1. Demodulación del patrón 1111 0000
    - 7.4.2. Demodulación del patrón 11 00
    - 7.4.3. Demodulación de otros patrones
  - 7.5. Respuesta del transmisor
- 8. Presupuesto
- 9. Conclusiones
  - 9.1. Conclusiones generales
  - 9.2. Futuras líneas de trabajo
- 10. Referencias bibliográficas

## 11. Anexos

# Índice de figuras

Figura 4.1: Distribución de cobertura celular.

Figura 4.2: Cobertura en entornos urbanos y rurales.

Figura 4.3: Estación móvil GSM

Figura 4.4: BTS en el sistema GSM

Figura 4.5: Arquitectura funcional del sistema GSM.

Figura 4.6: Ráfaga normal

Figura 4.7: Ráfaga corrección de frecuencia

Figura 4.8: Ráfaga de sincronización

Figura 4.9: Ráfaga de acceso.

Figura 4.10: Jerarquía de las tramas en GSM

Figura 5.1: Conexión entre dos transceptores USRP

Figura 5.2: NI USRP-2920.

Figura 5.3: Panel frontal del USRP-2920

Figura 5.4: Diagrama de bloques del NI USRP-2920

Figura 5.5: Agilent VSA 89600 (derecha), Agilent ESG E4438C (centro) y PC del laboratorio (izquierda).

Figura 5.6: PC portátil

Figura 5.7: Panel frontal de LabVIEW

Figura 5.8: Diagrama de bloques de LabVIEW

Figura 5.9: Ejemplo de uso del módulo MathScript RT

Figura 5.10: Interfaz gráfica del VSA

Figura 6.1: Variables de control del receptor

Figura 6.2: Interfaz gráfica del receptor



Figura 6.3: Resultados de la recepción de señales

Figura 6.4: Errores de recepción y transmisión

Figura 6.5: Esquema de la función *Extract Single Tone Information.vi*

Figura 6.6: Diagrama de bloques del módulo *CorregirOffsetFrec.vi*

Figura 6.7: Error de buffer del USRP

Figura 6.8: Configuración del número de muestras a adquirir.

Figura 6.9: Control del número de muestras adquiridas

Figura 6.10: Nueva adquisición de muestras del receptor

Figura 6.11: Recepción de tramas tras realizar sincronizar el sistema.

Figura 6.12: Recepción de trama desplazada en tiempo

Figura 6.14: Descomposición de la señal adquirida por el USRP.

Figura 6.15: Creación de un vector con un slot sin periodo de guarda

Figura 6.16: Diagrama de bloques del módulo *Eliminar Guarda.vi*

Figura 6.17: Selección del slot de tiempo a demodular

Figura 6.18: Configuración de los parámetros del sistema demodulador y del filtro Gaussiano.

Figura 6.19: Función *Demod Symbol Demap.vi* original y editada.

Figura 6.20: Salidas de la función *Demodulador.vi*

Figura 6.21: Diagrama de bloques del detector de patrones

Figura 6.22: Configuración del transmisor

Figura 6.23: Diagrama de bloques del transmisor desactivado

Figura 6.24: Diagrama de bloques de la respuesta del transmisor

Figura 7.1: Trama recibida no sincronizada en tiempo

Figura 7.2: Trama recibida sincronizada en tiempo

Figura 7.3: Trama GSM en el generador de señales con dos slots ocupados

Figura 7.4: Componentes IQ de dos slots sincronizados en tiempo

Figura 7.5: Configuración trama GSM en el generador de señales con ocho slots ocupados

Figura 7.6: Resultados de la recepción de ocho slots

Figura 7.7: Configuración de la trama GSM con un FCCH y dos TCH

Figura 7.8: Resultados de la recepción de tres slots.

Figura 7.9: Configuración trama con cuatro slots

Figura 7.10: Espectro de un FCCH y trama en el dominio del tiempo

Figura 7.11: Resultados tras nueva adquisición de muestras

Figura 7.12: *Breakpoint* tras ejecutar *Initiate.vi*

Figura 7.13: Corrección del desplazamiento en frecuencia

Figura 7.14: Dos slots con los periodos de guarda

Figura 7.15: Dos slots sin los periodos de guarda

Figura 7.16: Flujo de bits demodulados del patrón 0000 1111

Figura 7.17: Configuración del patrón 1100

Figura 7.18: Señal en tiempo del patrón 0011 y el FCCH

Figura 7.19: Flujo de bits demodulados del patrón 0011

Figura 7.20: Patrón 0011 de 128 bits

Figura 7.21: Señal recibida por el analizador vectorial. Un solo slot transmitido

Figura 7.22: Señal recibida por el analizador vectorial. Dos slots transmitidos

Figura 9.1: Inicio sesión USRP

Figura 9.2: Diagrama de bloques configuración señal

Figura 9.3: Diagrama de bloques configuración número muestras

Figura 9.4: Esquema función *Initiate.vi*.

Figura 9.5: Esquema función *Fetch Rx Data (poly).vi*

## Índice de tablas

Tabla 9.1: Costes de material

Tabla 9.2: Costes de personal

# Glosario

USRP	Universal Software Radio Peripheral
SDR	Software Defined Radio
PC	Personal Computer
NI	National Instruments
VI	Virtual Instruments
MIMO	Multiple-input Multiple-output
FM	Frecuencia Modulada
GSM	Global System for Mobile communications
ISM	Industrial, Scientific and Medical
MS	Mobile Station
BSS	Base Station Subsystem
NSS	Network Switching Subsystem
OSS	Operation Support Subsystem
BTS	Base Transceiver Station
BSC	Base Station Controller
MSC	Mobile Services Switching Center
HLR	Home Location Register
VLR	Visitor Location Register
AuC	Authentication Center
EIR	Equipment Identity Register
FDD	Frecuency Division Duplex
FDMA	Frecuency Division Multiple Access
TDMA	Time Division Multiple Access
NB	Normal Burst
FB	Frequency Correction Burst

SB	Synchronization Burst
AB	Access Burst
DB	Dummy Burst
TCH	Traffic Channel
BCH	Broadcast Control Channels
BCCH	Broadcast Common Control Channel
FCCH	Frequency Correction Channel
SCH	Synchronization Channel
CCCH	Common Control Channel
RACH	Random Access Channel
PCH	Paging Channel
AGCH	Access Grant Channel
SDCCH	Stand-Alone Dedicated Control Channel
SACCH	Slow Associated Control Channel
FACCH	Fast Associated Control Channel
IP	Internet Protocol
DDC	Digital down converter
CAD	Conversor analógico digital
CDA	Conversor digital analógico
SIM	Subscriber Identity Module
VSA	Vector Signal Analyzer
GMSK	Gaussian minimum shift keying
MSK	Minimum shift keying
CNAF	Cuadro nacional de atribución de frecuencias

# **1. Introducción**

## **1.1. Experiencias previas**

Este TFG se plantea como continuación de la implementación de la señalización de una estación base GSM realizada por el cotutor Borja Genovés. Dicho trabajo [1] es muestra del objetivo del departamento de Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid de realizar diversas implementaciones haciendo uso del SDR o radio definida por software.

## **1.2. Objetivos**

El objetivo principal de este trabajo es realizar el diseño e implementación de un procedimiento conjunto de recepción y transmisión de una estación base GSM. Este procedimiento consiste en la solicitud de transmisión por un móvil (usando un generador de señales), la recepción por parte de la estación base (realizada por un transceptor USRP), y la asignación de canales por parte de la estación base al móvil (realizada por un segundo transceptor USRP).

Dicha implementación debe ser realizada una plataforma SDR constituida por dos transceptores USRP y el entorno de desarrollo LabVIEW. La plataforma SDR permite realizar implementaciones en software de módulos que típicamente estaban implementados en hardware. Esto hace que sea una herramienta potente para el uso didáctico, siendo este otro de los objetivos del TFG. Se pretende que en un futuro se pueda proporcionar al alumno ciertos bloques del sistema implementado con el fin de que esté complete los módulos restantes.

## **1.3. Contenido de la memoria**

El trabajo está dividido en capítulos con el fin de facilitar al lector la consulta de la sección que le interese.

Capítulo 1: Se realiza una breve introducción sobre la implementación realizada.

Capítulo 2: Se plantea el problema a resolver por el trabajo.

Capítulo 3: Se muestra las fases del trabajo realizado.

Capítulo 4: Se realiza una descripción del estándar GSM.

Capítulo 5: Se describe el entorno de trabajo usado para este trabajo, incluyendo hardware y software.

Capítulo 6: Se detalla el diseño de los bloques implementados en el receptor y transmisor GSM.

Capítulo 7: Exposición y análisis de los resultados obtenidos.

Capítulo 8: Se detalla el presupuesto del trabajo.

Capítulo 9: Conclusiones del trabajo.

Capítulo 10: Se incluyen las referencias bibliográficas consultadas.

Capítulo 11: Se muestran los anexos del trabajo, que detallan el modo de funcionamiento de los transceptores USRP mediante LabVIEW.

## 2. Planteamiento del problema

### 2.1. Estado del arte

La rápida evolución de la electrónica digital en las últimas décadas ha hecho posible que procesos que usualmente se realizaban por medio de hardware puedan ser realizados usando software gestionado por un procesador.

Una plataforma SDR básica consta de un ordenador equipado con un conversor analógico digital y un adaptador de radio frecuencia. Una gran parte del procesamiento de las señales se realiza en procesador del computador, en lugar de utilizar hardware específico. Esto permite cambiar los protocolos y el procesamiento de la señal, simplemente modificando el software.

La tecnología SDR fue empleada por primera vez en las fuerzas armadas de EE.UU. en 1991 con el objetivo de usar un solo dispositivo radio que trabajara con diez protocolos de radio distintos y que pudiera operar en un amplio rango de frecuencias [2]. Tras eso el uso de las plataformas SDR se fue extendiendo.

La gran ventaja que aportan estas plataformas es la adaptabilidad de los sistemas implementados frente a nuevas innovaciones, lo que supone un ahorro en los costes económicos. Por ejemplo una plataforma SDR diseñada para el uso de comunicaciones móviles puede ser actualizada por software para que soporte futuros estándares de comunicación, sin realizar actualizaciones en el hardware.

#### 2.1.1. Alternativas

La plataforma SDR implementada hace uso del hardware y software proporcionados por National Instruments, el transceptor USRP 2920 y el entorno LabVIEW. Sin embargo hay otras plataformas SDR que se podrían haber usado para la realización de este trabajo. La principal de ellas es la plataforma llamada *GNU Radio*.

*GNU Radio* es un conjunto de herramientas de software libre y código abierto que permiten el procesamiento de señales para una plataforma SDR [3]. Estos códigos pueden ser usados por cualquier hardware transceptor de señales de radiofrecuencia, entre los que se encuentran los siguientes:

- Transceptores USRP de National Instruments
- Fairwaves UmTRX: hardware abierto que puede trabajar en un rango de frecuencias desde 300 MHz a 3,8 GHz.



- Funcube Dongle: dispositivo receptor de banda estrecha con un rango de trabajo entre 64 MHz a 1,7 GHz.

Estos dispositivos son solo parte de los muchos equipos con compatibilidad con GNU Radio [4]. Esto se debe al carácter abierto y libre de la plataforma. Por ello el desarrollo de implementaciones en esta plataforma se realiza con los lenguajes de programación Python y C++.

El uso de esta plataforma en vez del entorno de desarrollo LabVIEW ahorraría significativamente los costes del trabajo, incluso se podría sustituir los transceptores NI USRP-2920 por otros más baratos. Sin embargo este TFG se presenta como una continuación del trabajo desarrollado previamente por Borja Genovés y los objetivos a cumplir dejan claro que la implementación se debe realizar íntegramente con el entorno LabVIEW.

## **2.2. Marco regulador**

Todos los equipos de radiocomunicaciones usados en el laboratorio cumplen con lo establecido en el Reglamento de uso del dominio público radioeléctrico, aprobado mediante el Real Decreto 863/2008, respetando las limitaciones impuestas en cuanto a la potencia máxima permitida.

Los equipos usados en ningún momento superan los 10dBm de potencia transmitida, límite impuesto para el equipamiento de laboratorios. Las potencias transmitidas por los equipos son las siguientes:

- Generador de señales ESG E4438C: se ha transmitido una potencia máxima de 0dBm, es decir, 1 mW.
- Transceptor USRP: ha transmitido con una potencia máxima inferior a 10 dBm.

Además el decreto establece que los equipos de radio configurables por software no pueden ser puestos en el mercado si no funcionan en las condiciones de radio establecidas en el CNAF. Por tanto los equipos usados cumplen con la normativa.

### **3. Cronograma**

El desarrollo de este TFG se llevado a cabo entre septiembre de 2013 y septiembre de 2014. Las fases del trabajo realizado han sido las siguientes:

- Primera fase: Familiarización con el entorno de trabajo y el estándar GSM.
- Segunda fase: Diseño e implementación en LabVIEW
- Tercera fase: Prueba del sistema
- Cuarta fase: Realización de la memoria

#### **3.1. Primera fase**

Desarrollada entre septiembre de 2013 y febrero de 2014. Esta fase se ha dedicado al aprendizaje del entorno de desarrollo LabVIEW realizando tutoriales proporcionados por el fabricante. La dificultad de LabVIEW radica en acostumbrarse a la programación gráfica. Además se realizó la ardua tarea de familiarizarse con el complejo estándar GSM.

En esta fase también se llevó a cabo el aprendizaje de los transceptores USRP y de las implementaciones realizadas por el cotutor. Aquí se pudieron comprobar los errores que habría que solucionar en la siguiente fase, como el error de buffer de la recepción de datos y la pérdida de sincronización del receptor.

#### **3.2. Segunda fase**

Llevada a cabo entre marzo y julio del 2014. En ella se ha realizado la implementación de los distintos módulos del receptor y transmisor en LabVIEW. Entre las tareas llevadas a cabo se encuentran:

- Implementación del control de buffer.
- Corrección de la sincronización del sistema.
- Procesamiento de las muestras recibidas.
- Respuesta de la estación base.

#### **3.3. Tercera fase**

Realizada en julio de 2014. Ha consistido en las pruebas y optimizaciones del sistema implementado. En esta fase se han producido las capturas de pantalla incluidas en esta memoria.

### **3.4. Cuarta fase**

La realización de la memoria ha durado un mes. Había que poner en orden todo lo hecho durante los meses anteriores, explicando con detenimiento los resultados más importantes.

## 4. Estándar GSM

GSM (*Global System for Mobile communications*) es una tecnología celular y digital utilizada para la transmisión de servicios de voz y datos móviles.

### 4.1. Visión general

GSM es el nombre de un grupo de estandarización establecido en 1982 destinado a crear un estándar de telefonía móvil común en Europa. La necesidad de estandarizar la telefonía móvil surgió debido a que las primeras evoluciones en las telecomunicaciones celulares fueron desarrolladas sin una estandarización común. Ello provocó principalmente problemas de compatibilidad entre los sistemas.

Durante los primeros años tras 1982, se debatieron y probaron las principales características del sistema. Los criterios básicos que debía cumplir el nuevo estándar eran los siguientes:

- Buena calidad de voz
- Bajo coste de terminal y servicio
- Soporte para itinerancia internacional
- Capacidad de uso en terminales de mano
- Eficiencia espectral
- Compatibilidad con RDSI

Es importante destacar que GSM ofrece recomendaciones, no requisitos. Las especificaciones GSM definen las funciones y requisitos de la interfaz, pero no abordan el hardware. Gracias a esto, los operadores pueden instalar equipos de distintos fabricantes.

### 4.2. Especificaciones

#### 4.2.1. Bandas de frecuencia

En un principio se había previsto que GSM operaría en la banda de los 900 MHz. Sin embargo, en 1993 el operador británico Mercury One2One lanzó una red que operaba en una nueva banda de 1800 MHz, pero que esencialmente era la misma tecnología que GSM [5]. Tras esto nuevos operadores comenzaron a adoptar nuevas bandas de frecuencias. Paralelamente en EEUU se habilitó la banda de frecuencias de 1900 MHz para el uso de tecnología celular.

El uso de frecuencias superiores a la banda de 900 MHz provoca que la propagación de las señales sea menor, lo que se soluciona aumentando el

número de estaciones base. Finalmente las distintas bandas espectrales usadas fueron incluidas en el estándar GSM. Actualmente las bandas de frecuencia que operan son las siguientes:

- GSM 850: Empleada en EE.UU, Sudamérica y Asia.
- GSM 900: Banda de frecuencia original de Europa y extendida por todo el mundo.
- GSM 1800: Conocido como DCS 1800 (*Digital Cellular System 1800*)
- GSM 1900: Usada en América.

#### 4.2.2. Modulación

La modulación empleada en GSM es una variante de la modulación MSK (*Minimum Shift Keying*), llamada GMSK (*Gaussian Minimum Shift Keying*). Tiene la ventaja de usar un menor ancho de banda gracias a que la señal a modular es introducida en un filtro gaussiano antes de ser aplicada al circuito modulador, de manera que las transiciones de fase son suavizadas. Sin embargo esta modulación produce ISI (Interferencia Inter Simbólica) debido a que la duración del pulso a la salida del filtro supera el tiempo de un bit.

### 4.3. Arquitectura de la red

La estructura de la red GSM fue diseñada como respuesta a un problema: el limitado ancho de banda disponible para la telefonía móvil. La telefonía celular es la solución que GSM da a ese problema.

La telefonía celular consiste en dividir la superficie a la que se desea dar cobertura en pequeñas secciones llamadas células o celdas. La cobertura de cada célula es suministrada por una estación base o repetidor. Para evitar el problema de las interferencias entre células contiguas, se hace que cada estación base use canales distintos a los de las células vecinas.

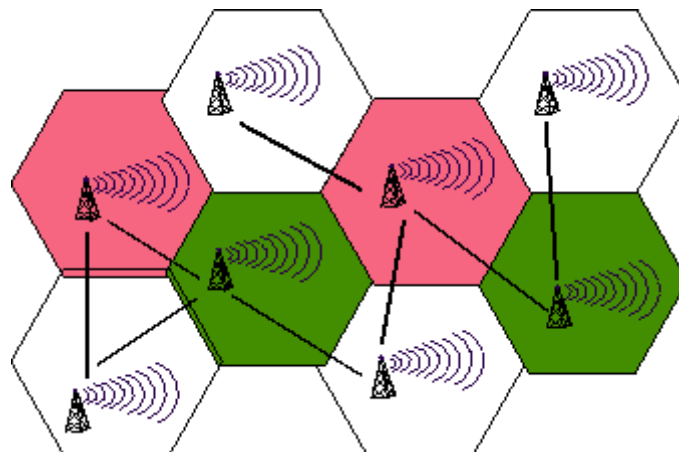


Figura 4.1: Distribución de cobertura celular [6].

La figura 4.1 sirve como ejemplo de la distribución de los canales en distintas celdas, en el que cada color representa un mismo canal. Se puede observar que ninguna célula está contigua a otra que use sus mismos canales de tráfico. Obviamente las señales de una estación base llegan hasta alguna celda que use los mismos canales, sin embargo la potencia de las señales será lo suficientemente baja como para interferir.

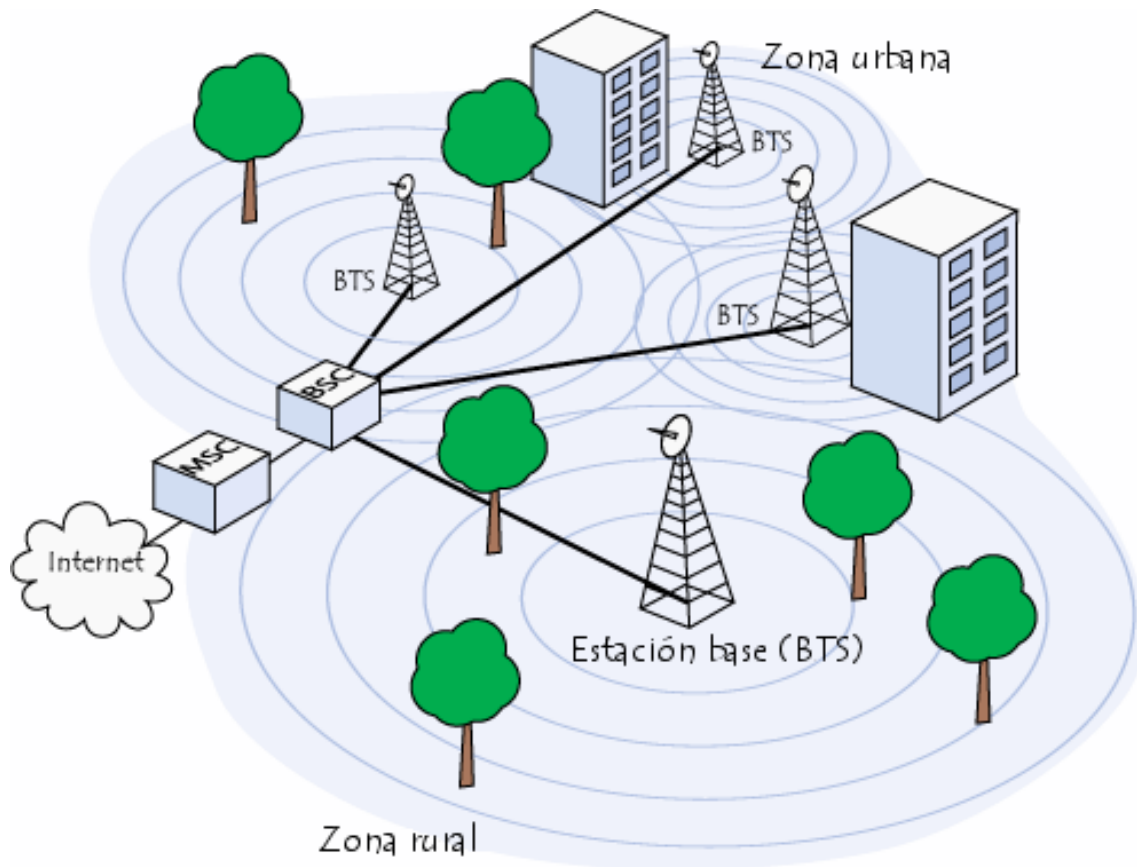


Figura 4.2: Cobertura en entornos urbanos y rurales [7].

El tamaño de una celda viene determinado por la cantidad de usuarios a las que debe dar servicio. Cuanto menor sea el radio de una celda, mayor será el ancho de banda disponible. En la práctica, las zonas urbanas poseen celdas de radios bajos, de pocos cientos de metros. Por otro lado en zonas rurales las celdas pueden llegar a tener un tamaño de varios kilómetros.

Funcionalmente, la arquitectura de la red GSM se encuentra dividida en una serie de entidades cuyas características están predefinidas. La separación de funcionalidades causa que la complejidad de la red sea menor. Estos subsistemas son:

- MS (*Mobile Station*).
- BSS (*Base Station Subsystem*).
- NSS (*Network Switching Subsystem*).
- OSS (*Operation Support Subsystem*).

#### **4.3.1. Mobile Station**

La estación móvil es la combinación de un terminal físico y una tarjeta SIM (*Subscriber Identity Module*). El terminal debe contar con un transceptor de radio, una pantalla y un procesador de señales digitales. La SIM almacena de forma segura la clave de servicio del suscriptor usada para identificarse ante la red.



Figura 4.3: Estación móvil GSM [8].

Las funciones de la MS son:

- Proporcionar al usuario la interfaz de comunicación con la red.
- Establecimiento de la conexión con la red.
- Sintonización de frecuencias y seguimiento de estaciones base.
- Transmisión y recepción de la información del usuario.
- Señalización en el interfaz radio
- Procesado de voz.

#### **4.3.2. Base Station Subsystem**

La BSS está compuesta por los BTS (*Base Transceiver Station*), pudiendo tener más de una estación, y los BSC (*Base Station Controller*). Las funciones de la BSS son:

- Transmisión y recepción radio.
- Localización de la MS.
- Establecimiento, control y finalización de las llamadas.
- Procesado de voz y adaptación de velocidades.

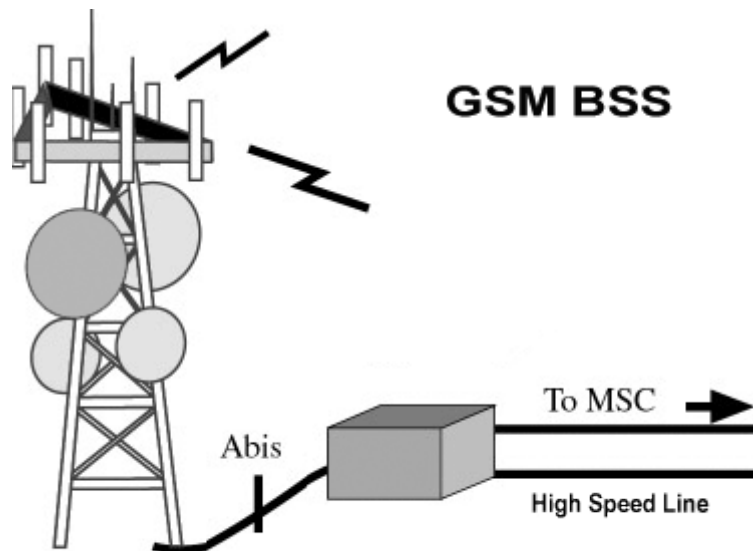


Figura 4.4: BTS en el sistema GSM [9].

La BTS corresponde a los transceptores y antenas usadas por cada célula de la red, en los que se realiza la transmisión y recepción de señales y su procesamiento digital. Usualmente se coloca en el centro de una célula y su potencia de transmisión define el tamaño de la célula. Cada BTS tiene entre 1 y 16 transceptores, dependiendo de la densidad de los usuarios en la célula.

El BSC se encarga de gestionar los recursos radio de varias BTS y conectar con la central de conmutación móvil. La gestión de los recursos incluye la configuración de los radiocanales, saltos de frecuencia y los traspasos.

#### **4.3.3. Network Switching Subsystem**

El subsistema de conmutación de red realiza la conmutación de llamadas entre los usuarios de la red móvil y otros fijo o móvil, así como la gestión de los servicios móviles. El NSS está formado por los siguientes elementos:

- **MSC (*Mobile Services Switching Center*)**: Es la parte principal del subsistema de red. Se encarga de la conmutación de llamadas entre los usuarios de redes móviles y otras redes, así como la gestión de los servicios móviles. Estos servicios son el registro, autenticación, actualización de la ubicación, los traspasos, y enrutamiento de llamadas.
- **HLR (*Home Location Register*)**: Es una base de datos que almacena y gestiona la información de los abonados permanentes. Cada operador tiene su propio HLR.



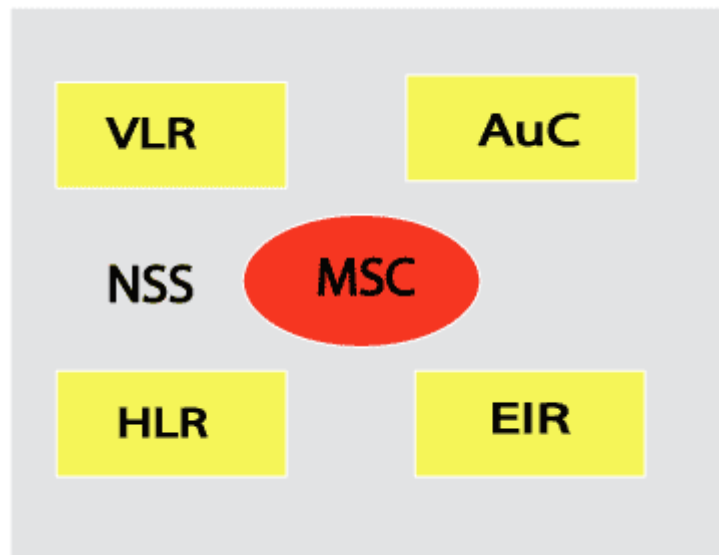


Figura 4.5: Arquitectura funcional del sistema GSM.

- VLR (*Visitor Location Register*): El VLR es una base de datos que contiene información temporal acerca de los suscriptores visitantes no registrados en el HLR del operador. El VLR está siempre integrado con el MSC.
- AuC (*Authentication Center*): El centro de autenticación es una base de datos protegida que almacena una copia de la clave secreta almacenada en la tarjeta SIM de cada abonado.
- EIR (*Equipment Identity Register*): es una base de datos que contiene una lista de todos los equipos móviles válidos en la red.

#### **4.3.4. Operation Support Subsystem**

El OSS es la entidad funcional desde la que el operador monitoriza y gestiona la red.

### **4.4. Capa física**

GSM es un sistema full-duplex, lo que quiere decir que tiene dos radiocanales por comunicación, permitiendo transmitir datos de forma simultánea en ambas direcciones. Para simplificar el diseño de los circuitos transmisores y receptores GSM utiliza FDD (*Frequency Division Duplex*), es decir, los canales ascendente (de móvil a estación base) y descendente (de estación base a móvil) tienen una separación mínima de 45 MHz.

#### **4.4.1. Tecnologías de acceso**

La interfaz radio de GSM combina el acceso por división en tiempo con el acceso por división de frecuencia. La tecnología de división de frecuencia

empleada es FDMA (*Frequency Division Multiple Access*) que divide el ancho de banda en múltiples radiocanales a los que asigna 200 kHz a cada uno.

TDMA (*Time Division Multiple Access*) es la tecnología de acceso en tiempo empleada. Consiste en dividir cada radiocanal en ocho intervalos de tiempo, denominados TN0 - TN7, que conforman una trama. Estos intervalos de tiempo tienen una duración de 0,577 ms, por tanto una trama presenta una duración de 4,615 ms.

El estándar GSM-900 trabaja en la banda de 890,2 MHz a 959,8 MHz, reservando para el canal ascendente la banda de 890,2 MHz a 914,8 MHz y para el canal descendente la banda de 935,2 MHz a 959,8 MHz. Dado que cada radiocanal de GSM tiene un ancho de banda de 200 kHz, se dispone de un total de 125 radiocanales en ambas direcciones. Cada radiocanal de frecuencia se divide en ocho slots de tiempo, por lo que se alcanza un total de 1000 canales de tráfico.

#### 4.4.2. Tipos de ráfagas

La información que se transmite en cada intervalo se envía en ráfagas. Cada ráfaga contiene 148 bits a los que se añaden 8.25 bits de guarda. Estos periodos de guarda son necesarios para que la estación móvil complete sus rampas de subida y bajada de potencia, además de disminuir las colisiones por multitrayecto en la estación base. Dado que cada slot dura 0.577ms la tasa de bits es de 270.833 kbps.

Hay distintos tipos de ráfagas dependiendo de los canales lógicos a transmitir:

- *Normal Burst* (NB)
- *Frequency Correction Burst* (FB)
- *Synchronization Burst* (SB)
- *Access Burst* (AB)
- *Dummy Burst* (DB)

##### 4.4.2.1. Normal Burst

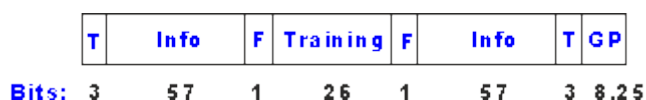


Figura 4.6: Ráfaga normal [9].

La ráfaga normal contiene 148 bits, de los cuales 114 son de información y 34 de señalización y control. Por tanto la eficiencia espectral es del 77%. Tanto el periodo de guarda como los bits de cola (en la figura: T) dan tiempo al receptor para completar sus rampas de subida y bajada de potencia. Los bits F indican el

tipo de mensaje transmitido. Los 26 bits centrales contienen la secuencia de entrenamiento, con los que se estima el canal y se ajusta el ecualizador.

Este tipo de ráfaga es usada tanto en el enlace ascendente como en el descendente y típicamente se utiliza para transmitir datos de voz digitalizados.

#### 4.4.2.2. Frequency Correction Burst



Figura 4.7: Ráfaga corrección de frecuencia [9].

La ráfaga de corrección de frecuencia transmite un tono puro desplazado 67,708 kHz de la frecuencia portadora, lo que permite al móvil sincronizar su reloj con el de la estación base. El tono puro está compuesto de una secuencia de 142 símbolos con valor '0' lógico.

Al igual que la ráfaga normal, se incluyen bits de cola y un periodo de guarda para dar tiempo a las bajadas y subidas de potencia en el receptor. Este tipo de ráfaga corresponde únicamente al enlace descendente.

#### 4.4.2.3. Synchronization Burst



Figura 4.8: Ráfaga de sincronización [9].

El propósito de este tipo de ráfaga es proveer sincronización a los móviles de la red. Para ello incluye una secuencia de entrenamiento más larga (64 bits) de lo habitual para que el móvil tenga mayor probabilidad de detectarla. Los 78 bits de información contienen los identificadores necesarios para realizar una completa sincronización del sistema.

Incluye bits de cola y periodo de guarda como las anteriores ráfagas y solo es usada en el enlace descendente.

#### 4.4.2.4. Access Burst

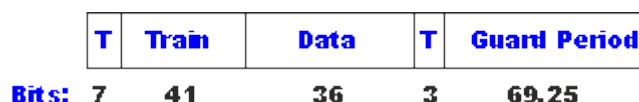


Figura 4.9: Ráfaga de acceso [9].

La ráfaga de acceso permite el acceso a la red y es usada únicamente en el enlace ascendente. Contiene más bits de cola y periodo de guarda que el resto de ráfagas, con el fin de evitar cualquier tipo de problema de temporización y

colisiones con otras ráfagas que puedan recibirse en el receptor en el mismo instante de tiempo.

#### 4.4.2.5. Dummy Burst

La ráfaga de relleno es empleada para simplificar la sincronización del sistema gracias a la poca potencia asociada que tiene. Debido a ello las ráfagas de sincronización y corrección de frecuencia son las de mayor potencia y el receptor del móvil las detecta con mayor facilidad.

#### 4.4.3. Canales lógicos

Un canal lógico es la información que compone un canal físico, es decir, una ráfaga. Estos se dividen en:

- Canales de tráfico
- Canales de control

##### 4.4.3.1. Canales de tráfico

Se encargan de transmitir tanto la voz como los datos del usuario y son portados en una ráfaga normal. Hay varios tipos, dependiendo del contenido y tasa de bit:

- *Traffic Channel/Full-rate Speech (TCH/FS)*: contiene voz digitalizada a una tasa de 13 Kbps.
- *Traffic Channel/Half-rate Speech (TCH/HS)*: contiene voz digitalizada a una tasa de 6,5 Kbps.
- *Traffic Channel/Full-rate Data (TCH/FD)*: transmite datos a una tasa de 9,6 Kbps.
- *Traffic Channel/Half-rate Speech (TCH/HD)*: transmite datos a una tasa de 4,8 Kbps.

##### 4.4.3.2. Canales de control

Los canales de control son usados principalmente para señalización y regulación de acceso al sistema. Hay varios tipos:

- Canales de control comunes: Se encargan de controlar el acceso de las estaciones móviles a la red.
  - *BCH (Broadcast Control Channels)*: son canales de enlace descendente y se encargan de difundir información de la estación base. Son:

- BCCH (*Broadcast Common Control Channel*): difunde información general de la estación base. Se transmite dentro de una ráfaga normal.
  - FCCH (*Frequency Correction Channel*): informa al móvil de la frecuencia portadora de la estación base. Se transmite dentro de una ráfaga de corrección de frecuencia.
  - SCH (*Synchronization Channel*): se encarga de sincronizar la estructura de las tramas. Incluida dentro de una ráfaga de sincronización.
- CCCH (*Common Control Channel*):
    - RACH (*Random Access Channel*): canal de subida por el que se realizan las peticiones de las estaciones móviles como el establecimiento de llamadas. Se transmite dentro de una ráfaga de acceso.
    - PCH (*Paging Channel*): Canal de bajada que notifica a la estación móvil de llamadas. Se transmite dentro de una ráfaga normal.
    - AGCH (*Access Grant Channel*): se encarga de asignar recursos a las estaciones móviles. Se transmite dentro de una ráfaga normal.
- Canales de control dedicados: canales bidireccionales que llevan señalización de la llamada y siempre se transmiten dentro de una ráfaga normal. Son los siguientes:
    - SDCCH (*Stand-Alone Dedicated Control Channel*): canal de intercambio de datos previo a un canal de tráfico.
    - SACCH (*Slow Associated Control Channel*): canal que coexiste con las llamadas y se encarga de mantener la señalización.
    - FACCH (*Fast Associated Control Channel*): canal de señalizaciones urgentes como traspaso de celda.

#### **4.4.4. Estructura de las tramas**

Como se ha dicho anteriormente, los intervalos de tiempo se agrupan de ocho en ocho formando tramas. Dependiendo de si los slots son para canales de tráfico o de control, estos se organizan en multitramas de distinto tamaño. Los intervalos de tiempo que contienen canales de tráfico son incluidos en multitramas de 26 tramas (M26), con una duración de 120 ms. Los slots con canales de control conforman las multitramas de 51 tramas (M51) y su duración es de 235,36 ms.

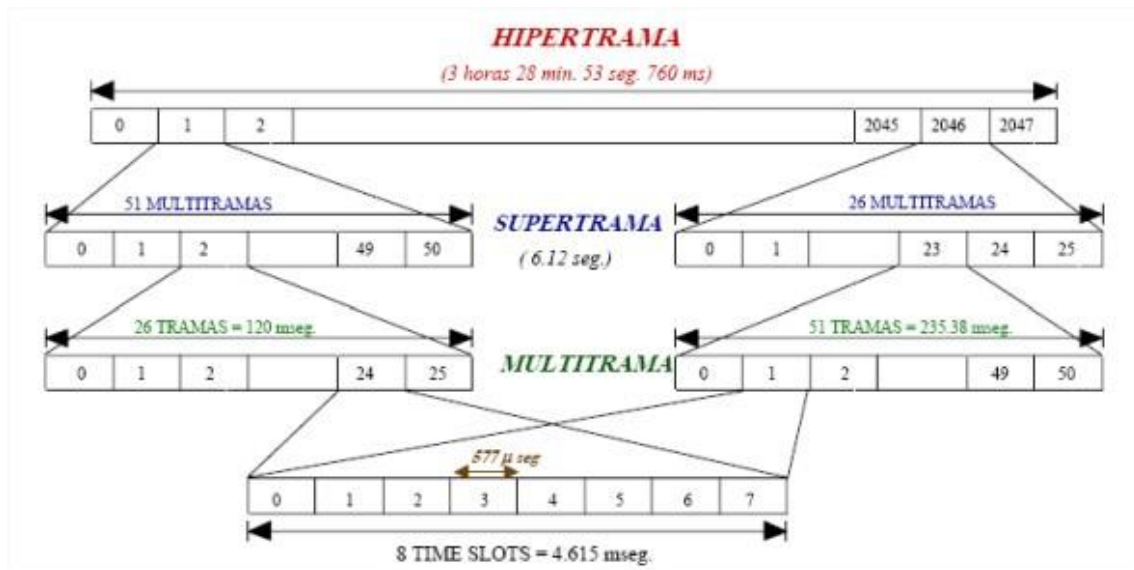


Figura 4.10: Jerarquía de las tramas en GSM [10].

Las multitramas conforman supertramas dependiendo del tipo de slots que contengan. 51 multitramas M26 conforman una supertrama y a su vez, 26 multitramas M51 forman otra supertrama, con la misma duración en tiempo, 6,12s.

Del mismo modo, una hipertrama está compuesta por 2048 supertramas. Una hipertrama es la estructura temporal de GSM y se repite cada 3 horas y 28 minutos aproximadamente.

## 5. ENTORNO DE TRABAJO

Este TFG se ha llevado a cabo en los laboratorios de comunicaciones móviles de la universidad Carlos III de Madrid, en el edificio Torres Quevedo del Campus de Leganés. Entre las herramientas usadas, en el laboratorio se encuentran las siguientes:

- Transceptores de radio NI URSP-2920
- Generador de señales Agilent ESG E4438C
- Analizador de señales vectoriales Agilent VSA 89600.
- PC

Además se ha hecho uso de un PC portátil personal para desarrollar este TFG.

Las dos principales herramientas utilizadas han sido los transceptores de radio NI URSP-2920 y el entorno de desarrollo LabVIEW, ambos suministrados por National Instruments. LabVIEW proporciona un sencillo modo de trabajar con el NI USRP gracias a su sencillez para desarrollar algoritmos de comunicaciones. Ambas herramientas componen una plataforma SDR basada en PC, orientada a la educación e investigación.

A continuación se va a presentar la instrumentación hardware y software utilizada en este trabajo:

### 5.1. Hardware

El hardware usado para el receptor GSM está compuesto por dos equipos NI USRP, uno para recepción y otro para transmisión, un cable MIMO para sincronización y datos que conecta ambos USRP y un cable Gigabit Ethernet para conectar uno de los USRP al PC. Ambos USRP realizan la función de estación base GSM o BTS. La figura 5.1 muestra la configuración usada: los dos USRP se encuentran conectados mediante el cable MIMO y uno de los USRP tiene la salida del cable Gigabit Ethernet.

Las señales GSM a recibir son generadas por el generador de señales vectoriales Agilent ESG E4438C, que realiza la función de estación móvil, y la respuesta del transmisor GSM implementado es analizada en el hardware Agilent VSA 89600.



Figura 5.1: Conexión entre dos transceptores USRP.

#### 5.1.1. NI USRP-2920

Los dispositivos usados para la recepción y transmisión de señales han sido los transceptores de radio programables NI USRP-2920. Este transceptor puede trabajar con frecuencias entre 50MHz y 2.2GHz, dispone de un ancho de banda base I/Q de hasta 20MHz y permite la transmisión de hasta 25MS/s.





Figura 5.2: NI USRP-2920 [11].

Cuando se conecta un NI USRP a un PC, este funciona como una SDR con capacidades de procesamiento de señales digitales. Los equipos NI USRP se comercializan pensando en el entorno educativo e investigador, siempre en combinación con el software LabVIEW.

Las principales características orientadas para los investigadores son el monitoreo espectral accesible, donde el NI USRP puede actuar como monitor espectral con hasta 40MHz de ancho de banda. Otra característica es la sencillez de implantación de nuevos protocolos de comunicación personalizados por el investigador. Por último se pueden implementar sistemas 2x2 MIMO gracias al cable MIMO para sincronización y datos [10].

Los USRP son muy útiles en el ámbito educativo ya que proporcionan una manera accesible de experimentar de manera práctica los conocimientos teóricos. Además el rango de frecuencias con las que puede trabajar el hardware permite acceder a señales del mundo real como radio FM, GSM, radar y bandas ISM.

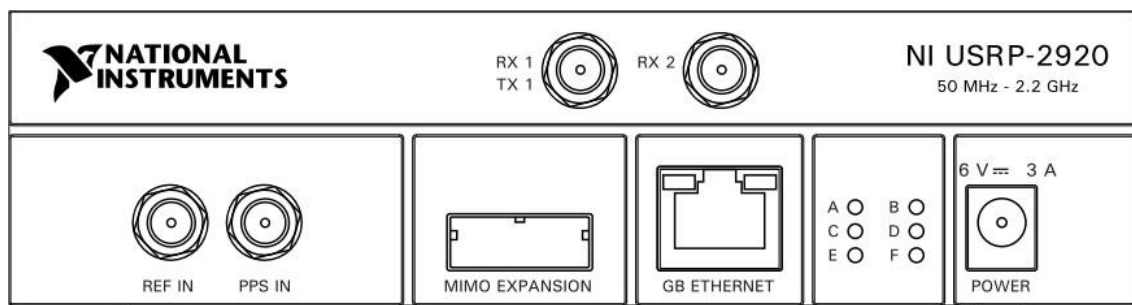


Figura 5.3: Panel frontal del USRP-2920 [11]

El panel frontal del USRP, mostrado en la figura 5.3, contiene las dos entradas para antenas: *TX1/RX1* y *RX2*, una entrada para el cable MIMO, un puerto Gigabit Ethernet y la entrada de la fuente de alimentación.

La siguiente figura muestra el diagrama de bloques del sistema destacando que el módulo de GPS (sección azul) no está disponible en el modelo NI USRP-2920.

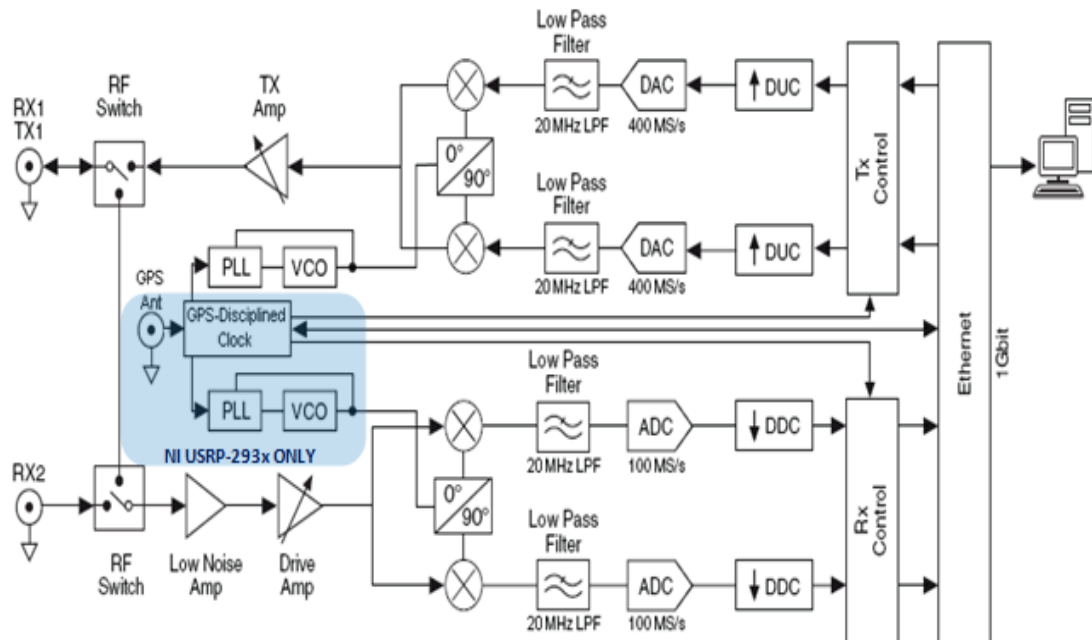


Figura 5.4: Diagrama de bloques del NI USRP-2920 [11]

#### 5.1.1.1. Receptor

El receptor del USRP consta primeramente de una etapa amplificadora que permite trabajar con ganancias entre 0 dB y 31,5 dB, con pasos de 0,5 dB. La máxima potencia de entrada permitida del receptor es de 0 dBm y la figura de ruido del amplificador es de entre 5 dB y 7 dB.

El receptor cuenta con dos canales, para las componentes I Q, con un filtro paso bajo de 20 MHz, un CAD que puede trabajar a 100 MS/s y un DDC que convierte la señal a banda base.

#### 5.1.1.2. Transmisor

La primera etapa del transmisor es un DUC que convierte la señal en banda base generada con LabVIEW en una señal portada en una frecuencia especificada. Después está el CDA que tiene una tasa máxima de muestreo de 400 MS/s y un filtro paso bajo de 20 MHz, ambos en dos canales para las componentes I Q.

La etapa amplificadora del transmisor tiene una ganancia entre 0 dB y 31 dB, en pasos de 1 dB. La potencia máxima transmitida va de 17 dBm a 20 dBm para un rango de frecuencias entre 50 MHz y 1,2 GHz.

### **5.1.2. Agilent ESG E4438C Vector Signal Generator**

El generador de señales vectoriales ESG E4438C es un equipo de alto rendimiento capaz de emitir señales en banda base y en radio frecuencia. Presenta las siguientes características: una frecuencia mínima de transmisión de 250 KHz, una frecuencia máxima de 6 GHz, una resolución de 0,01 Hz, una potencia mínima de transmisión de -136 dBm y potencia máxima de 16 dBm en la banda de trabajo.

Además este equipo es capaz de emitir señales GSM con cualquier tipo de configuración deseada.



Figura 5.5: Agilent VSA 89600 (derecha), Agilent ESG E4438C (centro) y PC del laboratorio (izquierda).

### **5.1.3. Agilent VSA 89600S**

El analizador de señales vectoriales desarrollado por Agilent es un dispositivo capaz de analizar el espectro con una alta resolución, demodular

señales y analizar las señales en el dominio del tiempo;ofreciendo un ancho de banda de hasta 36 MHz. Comercialmente es útil para el análisis de señales reales como GSM, radar y aplicaciones de ultrasonidos.

#### 5.1.4. PC portátil

El PC usado para el desarrollo de este trabajo ha sido un *HP Pavilion dv6*, con un procesador *Intel Core i5* de dos nucleos a 2,4 GHz y una memoria RAM de 4 GB. El sistema operativo usado ha sido *Windows 7 Proffesional* de 64 bits.

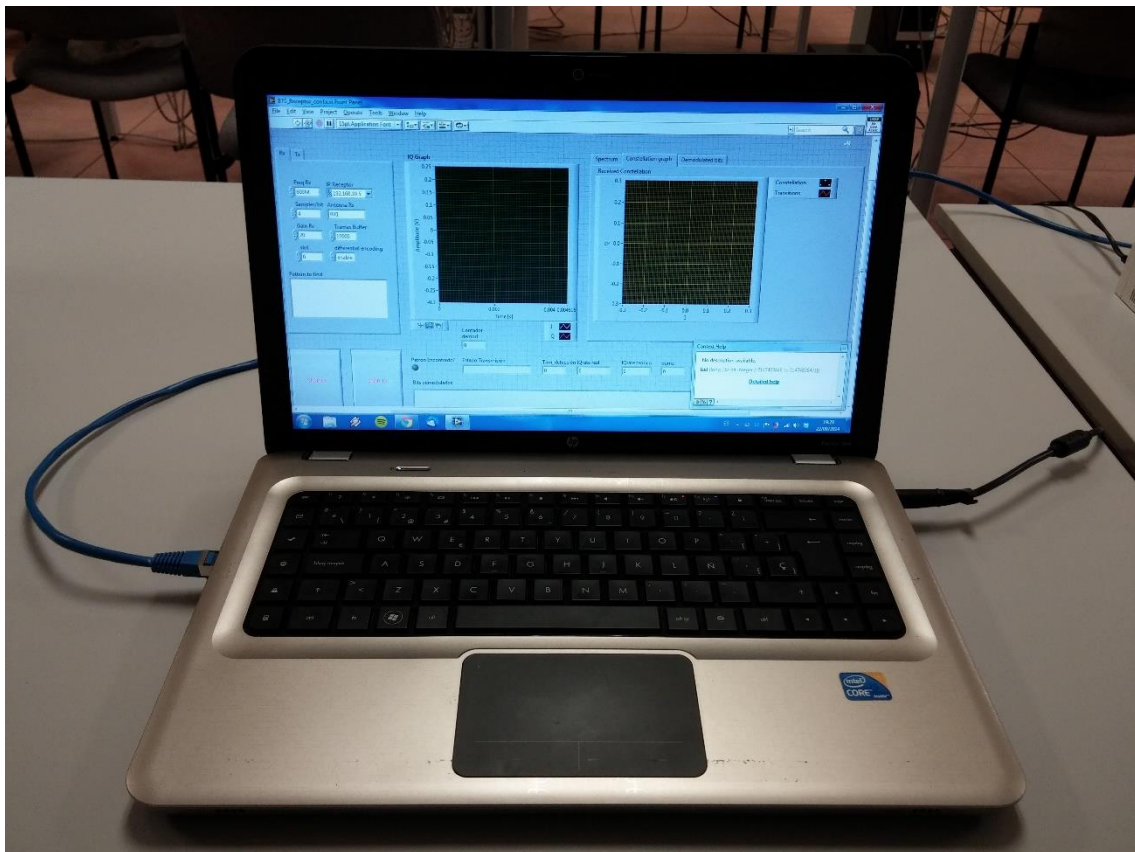


Figura 5.6: PC portátil.

## 5.2. Software

La implementación de este trabajo ha sido realizada usando el software de desarrollo LabVIEW 2010. Además se ha hecho uso del software VSA de Agilent para realizar la medida de las señales emitidas por el USRP de transmisión.

### 5.2.1. LabVIEW

LabVIEW es un entorno de desarrollo de National Instruments enfocado al diseño de sistemas de medidas y de control. Incrementa de manera notoria la

productividad gracias a la integración que ofrece entre software y hardware y a la abstracción de bajo nivel que proporciona,

Su principal característica es que es una plataforma de programación gráfica, lo que significa que no se escribe código como en un lenguaje de programación común. En lugar de eso, LabVIEW utiliza iconos para crear programas. Debido a esta característica, National Instruments recomienda el uso de un método de desarrollo de software que simplifica la solución de problemas. Este método de desarrollo de software de NI consta de cinco pasos:

- Definición del problema o escenario.
- Diseño de algoritmo y diagrama de flujo.
- Implementación del diseño
- Pruebas y verificación.
- Mantenimiento y actualización de la implementación.

El seguimiento de estos cinco pasos ayuda a los programadores en el desarrollo de programas al hacer que el código sea más legible, escalable y modificable, especialmente gracias a la sencillez de implementar los diagramas de flujo debido a la programación gráfica.

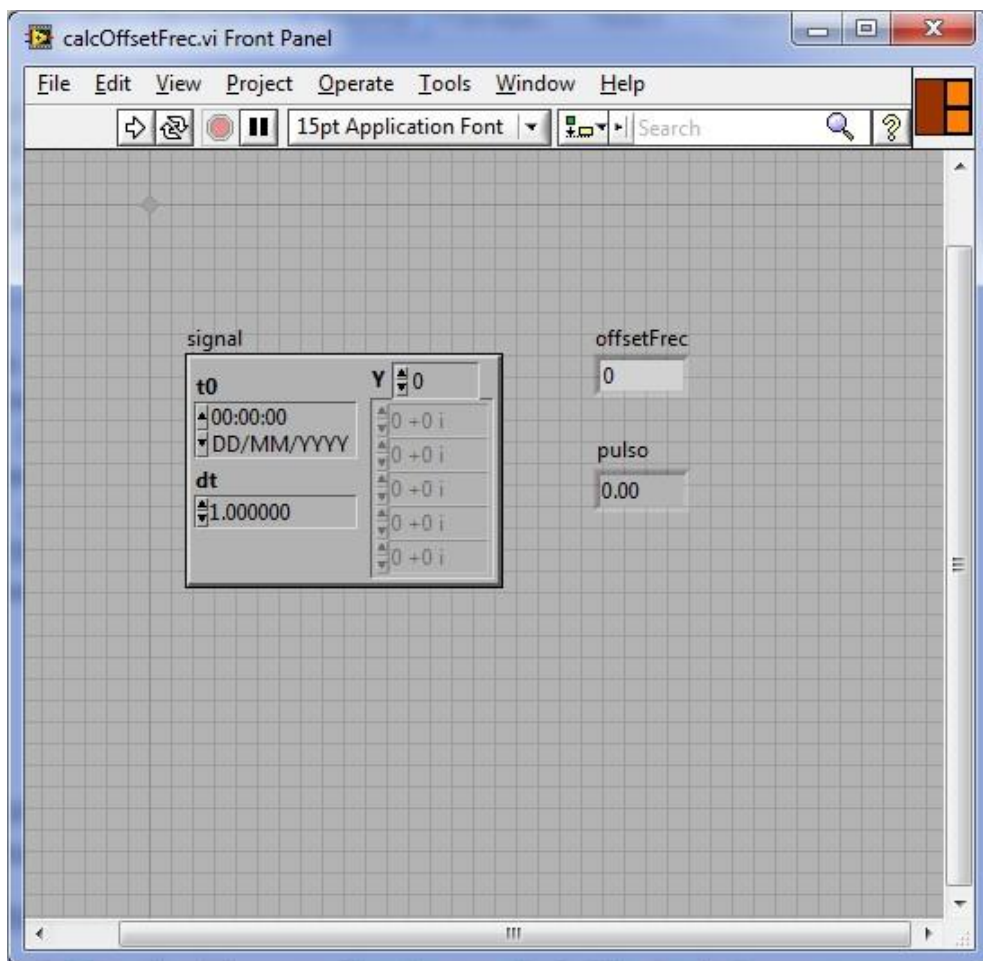


Figura 5.7: Panel frontal de LabVIEW.



Los programas creados en LabVIEW son llamados VI (*Virtual Instruments*) debido al origen que tiene la plataforma de trabajar con instrumentos como osciloscopios o multímetros. Cada VI consta de dos partes, el Panel Frontal y el Diagrama de Bloques. El Panel Frontal es la interfaz con la que el usuario interactúa con el VI, mediante controles, indicadores y entradas y salidas interactivas.

El Diagrama de Bloques es la ventana en la que el desarrollador añade código usando representaciones gráficas de funciones para controlar los objetos del Panel Frontal.

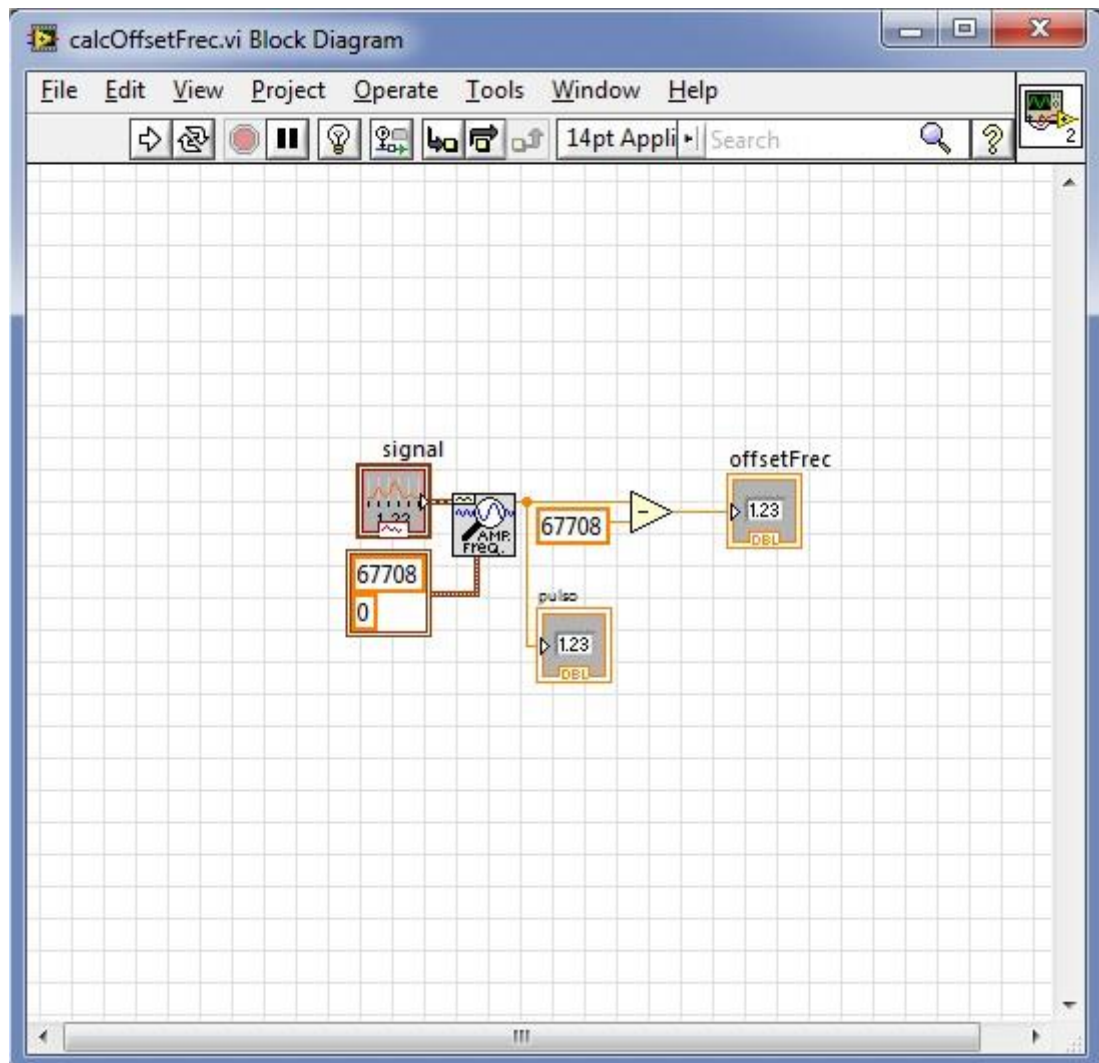


Figura 5.8: Diagrama de bloques de LabVIEW.

En los drivers de los USRP se añaden dos de las herramientas más importantes para la plataforma LabVIEW: el LabVIEW Modulation Toolkit y el LabVIEW MathScript RT Module.

#### 5.2.1.1. LabVIEW Modulation Toolkit

El LabVIEW Modulation Toolkit es una librería que añade VIs diseñados para los transceptores NI USRP que permiten el rápido desarrollo de sistemas

de comunicaciones, tanto simulados como reales. Incluyen también Vis para operar con señales en vivo. En las librerías se proporcionan también VIs para la generación de secuencias PN, la codificación de canal y la modulación de banda base. Para la recepción de señales se incluyen módulos para la demodulación, ecualización, decodificación de canal y más. En resumen, ese conjunto de herramientas habilita al usuario para desarrollar un sistema de comunicaciones por completo.

#### 5.2.1.2. LabVIEW MathScript RT Module

Este módulo permite integrar código de Matlab en los VIs, ejecutarlo y combinarlo con LabVIEW. La figura muestra un ejemplo de código .m en un Diagrama de Bloques de LabVIEW. Sin embargo, para la realización de este TFG no se ha hecho uso de este módulo ya que uno de los objetivos era la implementación completa en el entorno de desarrollo LabVIEW, y así favorecer a una ejecución más rápida del programa.

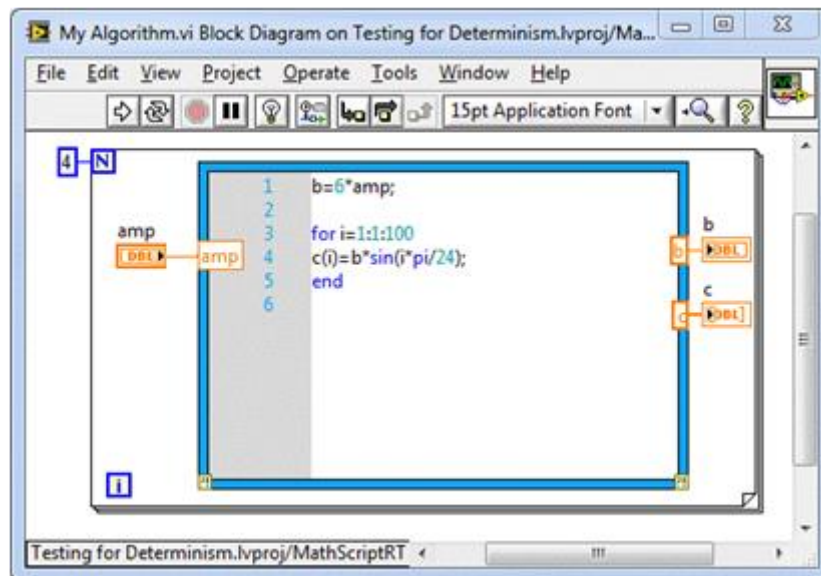


Figura 5.9: Ejemplo de uso del módulo MathScript RT.

#### 5.2.2. Software VSA

El software VSA (*Vector Signal Analysis*) es la herramienta que en combinación con el equipo Agilent 89600S permite el análisis de las señales GSM emitidas por el transmisor USRP.

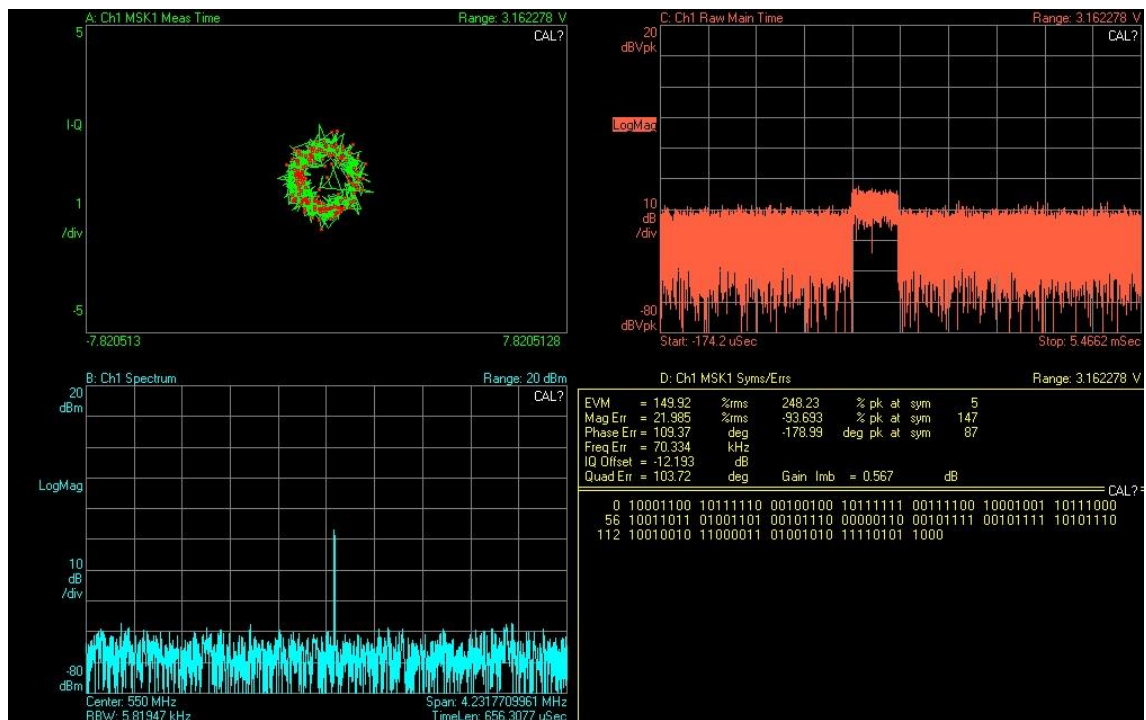


Figura 5.10: Interfaz gráfica del VSA.

En la figura 5.10 se muestra la interfaz del analizador vectorial usada para medir las señales GSM. Los recuadros superiores representan la constelación recibida y la señal en el dominio del tiempo (izquierda y derecha de la imagen). En la parte inferior se puede ver el espectro en frecuencia de la señal recibida y a la derecha la demodulación de bits realizada.



## 6. Diseño

El objetivo inicial de este TFG es realizar un receptor de una BTS que sea capaz de detectar una solicitud de comunicación de una estación móvil y que posteriormente responda con la asignación de canales al terminal móvil. Sin embargo se ha diseñado el receptor de manera que sea capaz de recibir la señal del móvil, detectar un patrón de bits preestablecido, patrón que es emitido desde el generador de señales en uno de los slots de tiempo, y finalmente realizar una transmisión de respuesta al móvil.

Se ha continuado el diseño básico del receptor realizado por el cotutor Borja Genovés, en el cual se realizaba una primera sincronización en tiempo detectando el canal FCCH.

Por tanto los módulos implementados son los siguientes:

- Interfaz de usuario: contiene las variables de control para gestionar el sistema.
- Ajuste de frecuencia: necesario para corregir el error de frecuencia de los equipos usados.
- Control de buffer: requerido para que la ejecución del programa no se interrumpa.
- Sincronización en tiempo periódica: necesaria para mantener el sistema sincronizado.
- Procesamiento de datos: en el que se adquieren las muestras recibidas, se demodula la señal y se realiza la detección de un patrón de bits.
- Transmisor: donde se gestiona la transmisión de la BTS en función de la señal recibida.

### Consideraciones previas

Debido a las limitaciones técnicas de los transceptores USRP no es posible realizar un barrido de toda la banda de trabajo para detectar la frecuencia portadora de la señal a recibir. Por tanto, la frecuencia portadora es establecida manualmente. La frecuencia a la que emite la estación móvil, en este caso el generador vectorial de señales, es de 600 MHz; la frecuencia de transmisión de la estación base ha sido establecida en 645 MHz, es decir, 45 MHz por encima de la frecuencia del enlace de subida, como se establece en el estándar.

El anexo 2 incluido en el capítulo 9 de este trabajo detalla cómo se realiza la adquisición de datos con el USRP.

## 6.1. Interfaz de usuario

El VI *Interfaz usuario.vi* es el menú principal del sistema implementado. En él se encuentran todas las variables de control necesarias para el funcionamiento del sistema y se muestran los resultados obtenidos.

En este VI además se realiza la ejecución de todo el sistema implementado. Aquí se llevan a cabo las llamadas al VI *Sync Tiempo.vi*, que realiza la sincronización del receptor, al VI *Receptor.vi*, que controla la recepción de muestras, la detección del patrón deseado, y la ejecución del transmisor.

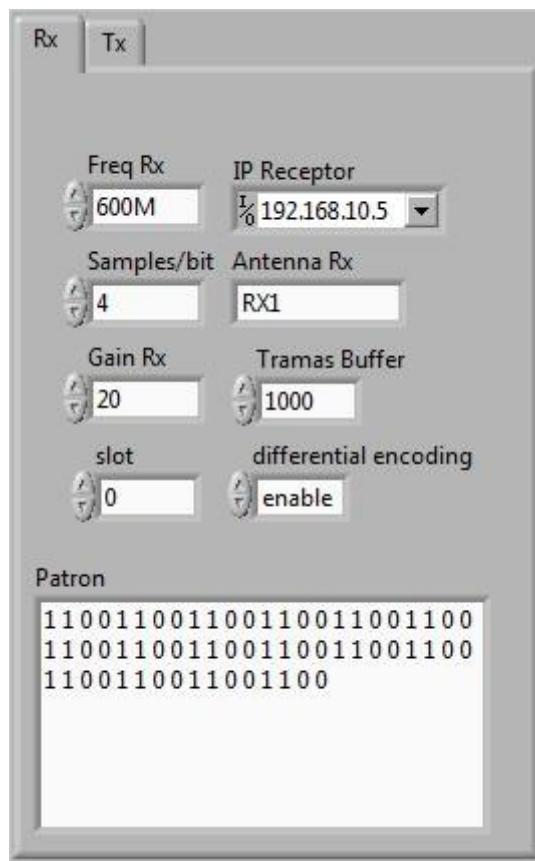


Figura 6.1: Variables de control del receptor.

Las variables de control que gestionan el receptor son las siguientes:

- *Freq Rx*: frecuencia portadora de recepción. Establecida en 600 MHz.
- *IP Receptor*: es la dirección IP correspondiente al USRP usado como receptor.
- *Samples/bit*: es el número de muestras por bit que debe adquirir el receptor. Establecido en 4.
- *Antenna Rx*: establece la antena del USRP que se desea usar para la recepción. Puede ser Rx1 o Rx2, dado que el USRP posee dos entradas para antenas.

- *Gain Rx*: la ganancia que aplicar a la señal recibida, expresada en dB. En este trabajo se ha usado una ganancia de 10 dB.
- *Tramas buffer*: es el número de tramas GSM que se adquiere en cada recepción de muestras. Establecido en 1000.
- *Slot*: el intervalo de tiempo que se desea demodular.
- *Differential encoding*: activa o desactiva la codificación diferencial.
- *Patrón*: el patrón que se desea detectar en la señal demodulada.

Las variables de control que gestionan el transmisor se detallan en el apartado 6.6. de este capítulo.

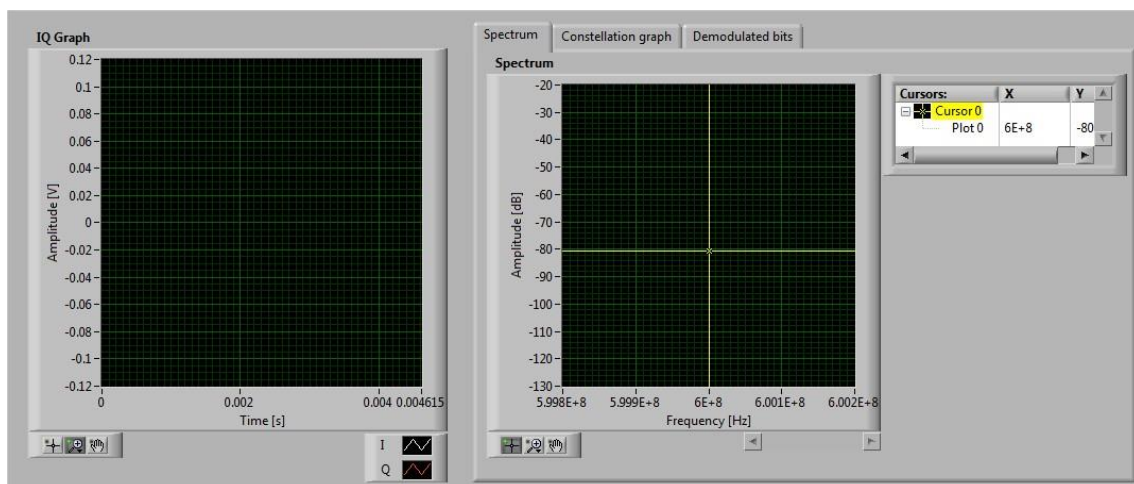


Figura 6.2: Interfaz gráfica del receptor.

La interfaz gráfica muestra los siguientes resultados de la recepción de señales:

- *IQ Graph*: gráfico que muestra las componentes I Q de la señal recibida en función del tiempo. El intervalo de tiempo mostrado es el equivalente a la duración de una trama GSM.
- *Spectrum*: gráfico que muestra el espectro de frecuencia de un slot de la señal recibida. El slot de tiempo seleccionado es el que dicte la variable de control *slot*.
- *Constellation graph*: la constelación recibida.
- *Demodulated bits*: gráfico que muestra los bits demodulados en función del tiempo.

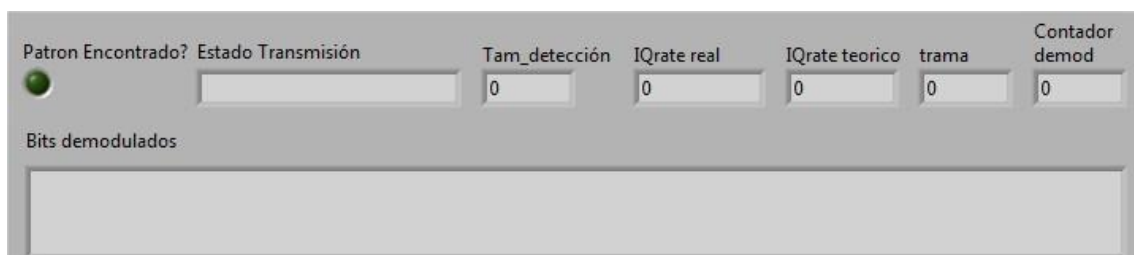


Figura 6.3: Resultados de la recepción de señales.

- *Patrón Encontrado?*: es un indicador que avisa si el patrón ha sido detectado.
- *Estado Transmisión*: que indica si el transmisor ha respondido a la estación móvil.
- *IQrate real*: es la tasa de bit con la que las muestras son adquiridas.
- *IQrate teórico*: es el valor teórico que debería tener la tasa de bit de adquisición.
- *trama*: indica el número de tramas analizadas por el receptor.
- *Contador demod*: indica las veces que el patrón ha sido detectado.
- *Bits demodulados*: muestra el flujo de bits demodulados.

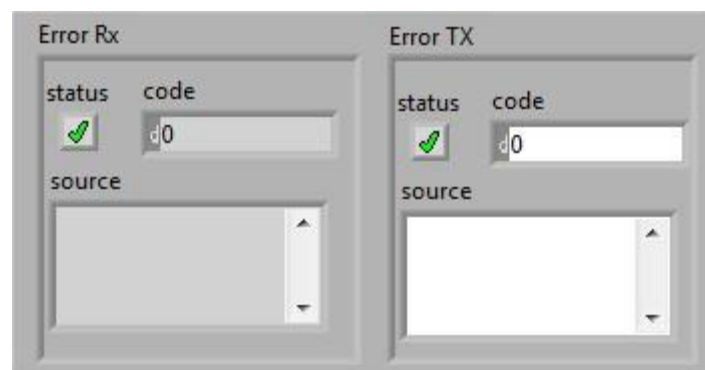


Figura 6.4: Errores de recepción y transmisión.

- *Error Rx*: muestra los errores surgidos durante la recepción. Se indica el estado del error, su identificador y la descripción.
- *Error Tx*: muestra los errores surgidos durante la transmisión.

Para obtener el IQ rate teórico se debe multiplicar la tasa de bit, que en GSM es de 270.833kbps, por el número de muestras por bit que se desea tomar. Para esto hay que tener en cuenta que para poder trabajar posteriormente con el periodo de guarda de cada slot, que es de 8.25 bits, se tiene que usar una tasa de muestreo tal que permita eliminar los 0.25 bits sobrantes. Si se adquieren 4 muestras por bit, los 0.25 bits corresponderán a una sola muestra. Por lo tanto la tasa IQ nos da el siguiente resultado:

$$\frac{\left(\frac{1625000}{6}\right)b}{s} * \frac{4S}{b} = \frac{1,083333MS}{s}$$

## 6.2. Sincronización del sistema

Antes de iniciar la recepción continua de datos, el receptor debe estar sincronizado en tiempo y ajustado en frecuencia con la estación móvil, es decir, con el generador de señales. Para ello se realizan los siguientes módulos:

- Sincronización en tiempo
- Ajuste de frecuencia

En GSM la sincronización en frecuencia se realiza haciendo uso de los canales lógicos FCCH y SCH, sin embargo, este trabajo ha hecho uso del módulo de sincronización en tiempo desarrollado por el cotutor Borja Genovés. Este módulo se describe brevemente en el siguiente apartado.

### **6.2.1. Sincronización en tiempo**

Cuando se inicia una comunicación GSM, el receptor debe detectar el FCCH enviado por la estación base para reconocer el canal a usar. El FCCH emite un tono único 67,7 kHz por encima de la frecuencia portadora. Este tono se utiliza para sincronizar el reloj local del receptor móvil con la estación base, lo que es necesario para extraer los datos a continuación.

El FCCH consta de una secuencia de todo ceros, secuencia a la que se aplica codificación diferencial y modulación GMSK. Posteriormente el FCCH es transmitido como una señal que combina las componentes I y Q con una diferencia de fase de 90°.

Este módulo está basado en la patente WO 91/10305, propiedad de SONY CORPORATION y SONY UNITED KINGDOM LIMITED, que realiza una sincronización en tiempo con el FCCH. Dicha patente describe un método que consiste en muestrear la señal en banda base y a continuación ir midiendo el incremento de fase entre dos muestras consecutivas de la señal muestreada.

El objetivo es calcular la correlación entre el flujo de datos muestreado y una señal de referencia generada por el receptor. Esta señal de referencia predefinida equivale a las muestras de un FCCH original. La señal entera se recorre recogiendo esos aumentos de fase, los cuales se comparan con un punto intermedio entre 0° y 180° y la suma de todas las diferencias se añaden a un vector. Además se realiza una ecualización del canal.

Finalmente, el valor mínimo del vector es detectado como el punto en el que comienza el slot de tiempo que contiene el FCCH.

### **Implementación**

Una vez que se inicia la sesión del USRP receptor, se adquieren un número determinado de muestras que son almacenadas en el buffer de recepción. Entonces se recogen las primeras muestras para realizar la sincronización en tiempo. El número de muestras que analizar tiene que ser tal que incluya un FCCH completo, por lo tanto, para asegurar la presencia completa de un FCCH se adquieren nueve slots de tiempo.

El número de muestras que componen un slot viene determinado por la multiplicación de la duración en tiempo de un slot, 0,577ms, por la tasa de adquisición de muestras del receptor.

Una vez adquiridas las muestras necesarias se calcula la correlación entre el flujo de datos recibido y un FCCH generado previamente manualmente.

El VI que realiza este módulo es el *Sync Tiempo.vi*. Tras realizar los cálculos y detectar el punto de inicio del FCCH, se adquieren y desechan las muestras que falten hasta completar una trama. De este modo la siguiente adquisición de datos comienza en el inicio de la siguiente trama.

### 6.2.2. Ajuste de frecuencia

Los equipos usados para transmitir y recibir las señales pueden introducir un pequeño error en la frecuencia de trabajo. Este desplazamiento de la frecuencia suele ser de pocas centenas de hertzios, pero suficientes para provocar un error en el receptor. Una vez que se realiza la sincronización en tiempo con el FCCH recibido, se ha de realizar la corrección de la frecuencia.

Para ello se ha de detectar la frecuencia a la que está la sinusoide transmitida en el FCCH, que según el estándar debe ser igual a una frecuencia 67,708 kHz superior a la frecuencia portadora. Una vez detectada la frecuencia, hay que compararla con la frecuencia teórica para hallar el desplazamiento provocado por el equipo de transmisión o el de recepción. La frecuencia teórica es conocida debido a que la frecuencia portadora de la señal es introducida manualmente.

Para corregir el desplazamiento en frecuencia se hace uso de las propiedades de la transformada discreta de Fourier (DFT). Se usa la DFT ya que se trabaja con muestras de secuencias discretas y finitas generadas a partir del muestreo de señales en tiempo. El teorema del desplazamiento dicta que multiplicando las muestras de la señal en tiempo por una fase lineal, las muestras en el dominio de la frecuencia sufrirán un desplazamiento circular. La siguiente figura muestra la propiedad y la equivalencia entre el dominio del tiempo y la frecuencia.

$$F \left( \left\{ x_n * e^{\frac{2\pi i n m}{N}} \right\} \right)_k = X_{k-m}$$

Donde  $i$  es la unidad imaginaria;  $n$  el intervalo de tiempo en segundos entre dos muestras consecutivas;  $m$  es el desplazamiento en Hz y  $N$  el número de muestras de un slot. El coeficiente  $x_n$  hace referencia a la secuencia en tiempo y  $X_{k-m}$  corresponde a la secuencia en frecuencia.

## Implementación

Una vez que el receptor se ha sincronizado en tiempo detectando el FCCH transmitido, analiza el slot correspondiente a dicho FCCH para calcular la frecuencia de su senoide. El offset en frecuencia se calcula haciendo uso de la función *Extract Single Tone Information.vi* incluida en las librerías de LabVIEW. Dicha función lee las muestras del FCCH y detecta la frecuencia del tono con mayor amplitud. Posteriormente compara la frecuencia de ese tono con los 67,708 kHz para hallar el desplazamiento en frecuencia del sistema.

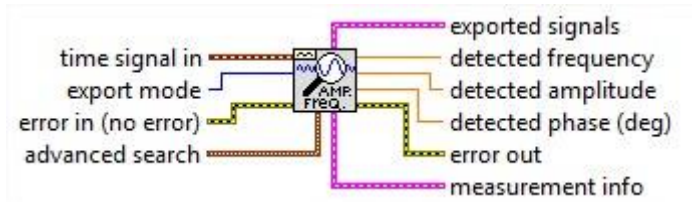


Figura 6.5: Esquema de la función *Extract Single Tone Information.vi*

La corrección de frecuencia se realiza en el sub VI *CorregirOffsetFrec.vi*, invocado después de realizar la sincronización en tiempo en el VI *Sync Tiempo.vi*. Este VI ha sido creado para corregir la frecuencia de cualquier señal que se introduzca como entrada. Esta función recibe como parámetros la señal a corregir, el offset en frecuencia y el número de muestras de un slot. El VI multiplica la señal recibida por la exponencial compleja y devuelve a la salida la señal con la frecuencia modificada.

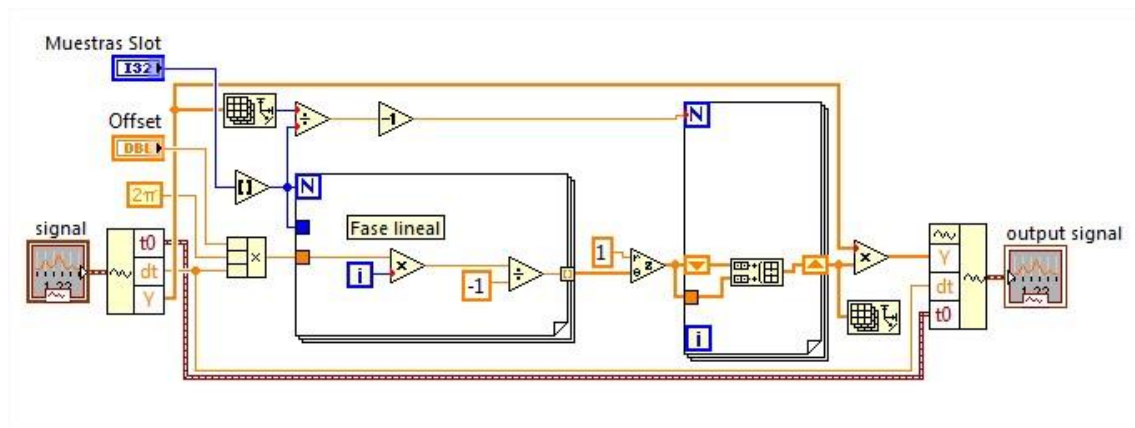


Figura 6.6: Diagrama de bloques del módulo *CorregirOffsetFrec.vi*

### 6.3. Control del buffer de recepción

Uno de los grandes problemas del sistema implementado es la imposibilidad de realizar una recepción de la señal de manera continua y sin pérdida de datos. Esto es debido a que el PC no procesa de manera instantánea las muestras adquiridas, por tanto estos datos se van acumulando en un buffer

de recepción. Pasado un tiempo este buffer llega a su límite, lo que causa que la ejecución del programa se interrumpa.

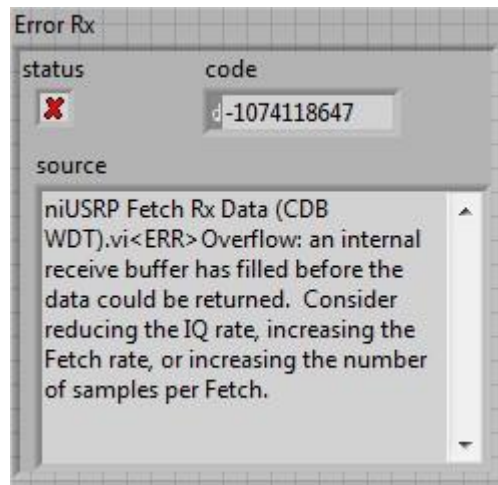


Figura 6.7: Error de buffer del USRP

La figura muestra el error mostrado después de que el receptor estuviera funcionando aproximadamente 30 segundos.

Como consecuencia, se ha decidido realizar una adquisición de datos controlada, para que el buffer de recepción no se llene. Para que el receptor pueda detectar rápidamente cambios en la señal emitida, es conveniente realizar una adquisición de pocas tramas. Adquirir pocas tramas conlleva que el tiempo de procesamiento sea menor y que la nueva adquisición de datos se realice antes.

Una alternativa a esta solución sería almacenar tantos datos como espacio haya en el buffer. Sin embargo el tiempo de procesamiento de todas las muestras sería muy elevado y la nueva adquisición de datos se realizaría mucho tiempo después, lo que disminuiría la sensibilidad del receptor ante posibles cambios en los datos enviados.

### Implementación

Para implementar la solución en LabVIEW se ha optado por usar una variable de control llamada *Tramas Buffer* que establece el número de tramas GSM que se quiere recibir. Por defecto el valor es de 1000 tramas, pudiéndose aumentar o disminuir dependiendo de la capacidad de procesamiento del ordenador en el que se ejecuta.



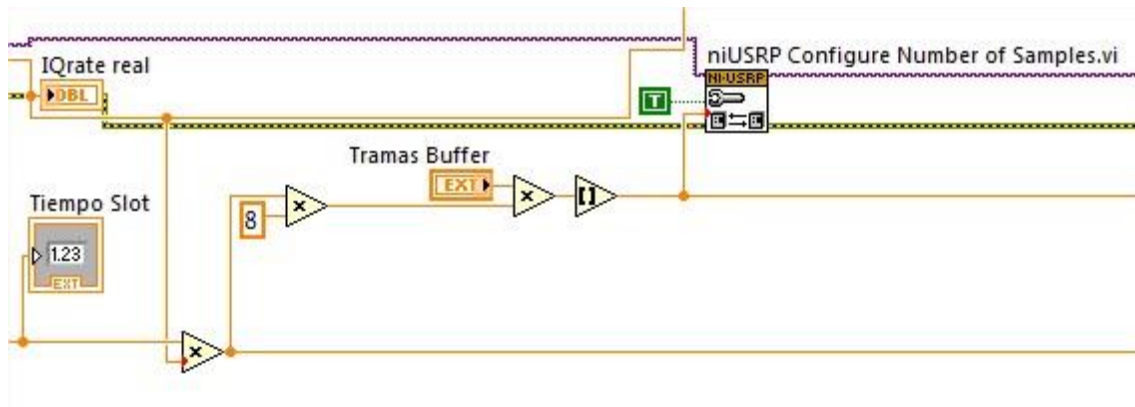


Figura 6.8: Configuración del número de muestras a adquirir.

En primer lugar se calcula el número de muestras de una trama multiplicando por ocho las muestras de un slot. Ese dato es multiplicado por la variable de control *Tramas Buffer* y el resultado se une a la entrada *number of samples* del VI.

Una vez configurado el número de muestras, se crea una variable llamada *Cont Muestras* que se encarga de contar las muestras ya leídas del buffer. Este contador se actualiza durante toda la ejecución del programa, cada vez que se ejecuta el vi *niUSRP Fetch Rx Data.vi* para recoger muestras del buffer.

Para que la ejecución del programa no se interrumpa hay que evitar que el buffer de recepción se vacíe. Cuando el contador de muestras se acerca al número de muestras recibidas se realiza una lectura de las muestras restantes en el buffer y se ejecuta de nuevo la recepción de muestras. Como muestra la siguiente figura, mediante un comparador se ha establecido que cuando queden menos de 16 slots, es decir dos tramas, por leer del buffer, se devuelva *True*.

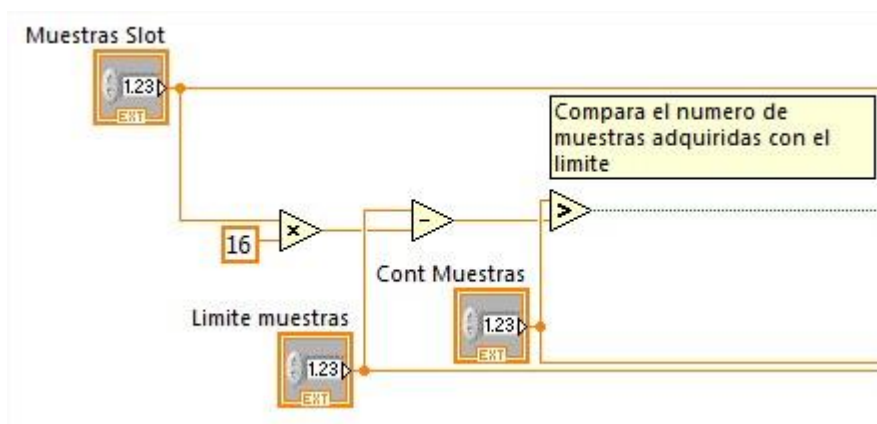


Figura 6.9: Control del número de muestras adquiridas.

Esa variable booleana actúa de control en la siguiente estructura de control de casos, similar a una estructura de control IF de los lenguajes de programación comunes. Cuando su estado es verdadero, se realiza la lectura de las muestras restantes y acto seguido se ejecuta de nuevo la recepción de muestras. Por lo

tanto, después de ejecutar la estructura, el receptor tiene el buffer lleno de muestras con las que seguir trabajando.

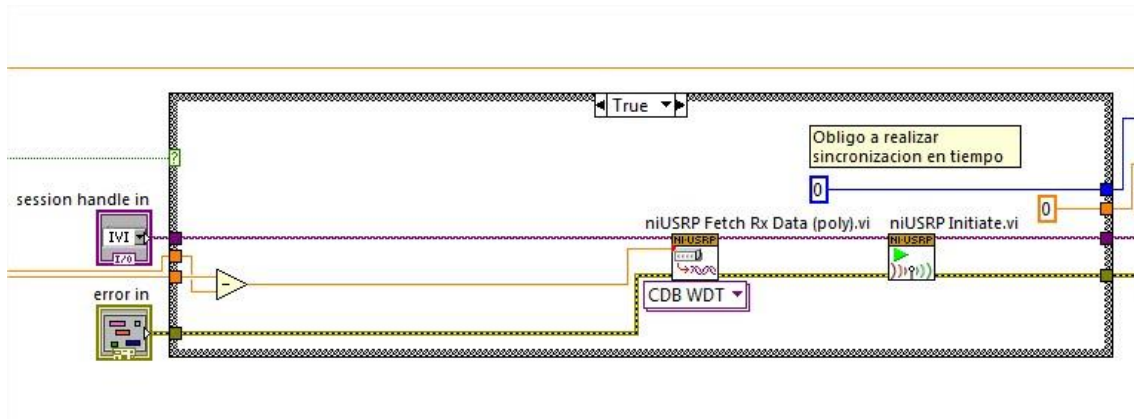


Figura 6.10: Nueva adquisición de muestras del receptor.

Al realizar una nueva adquisición de muestras, la sincronización en tiempo del sistema se pierde por lo que es necesario detectar de nuevo un FCCH para sincronizar el receptor. Por ello se devuelve el valor 0 en una de las salidas de la estructura. Este valor es la entrada del siguiente bloque del receptor, que se encarga de mantener la sincronización en tiempo del sistema.

## 6.4. Mantenimiento de la sincronización

Una vez que el sistema está sincronizado en tiempo y frecuencia mediante los métodos descritos anteriormente, el receptor es capaz de recibir tramas de manera continua. La primera adquisición de tramas debería tener la forma de la figura 6.11.

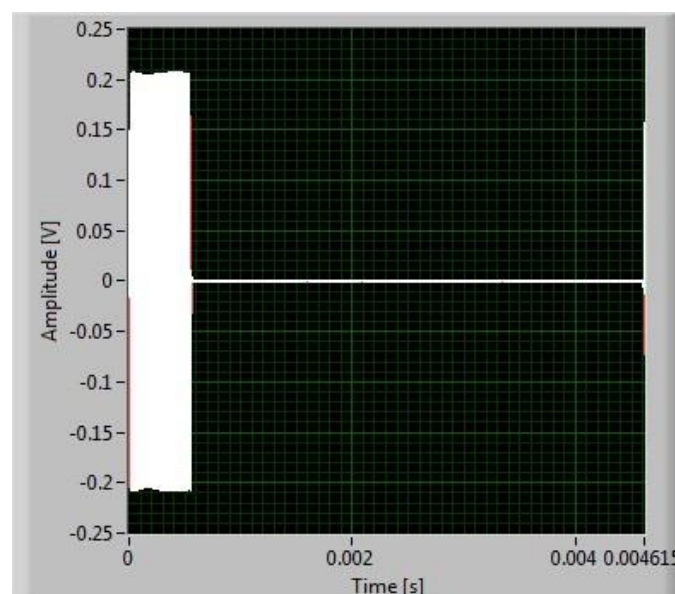


Figura 6.11: Recepción de tramas tras realizar sincronizar el sistema.

Sin embargo tras leer un número muy bajo de tramas, el receptor comienza a perder la sincronización en tiempo. La figura 6.12 muestra el desplazamiento en tiempo de la señal recibida tras un corto periodo de tiempo.

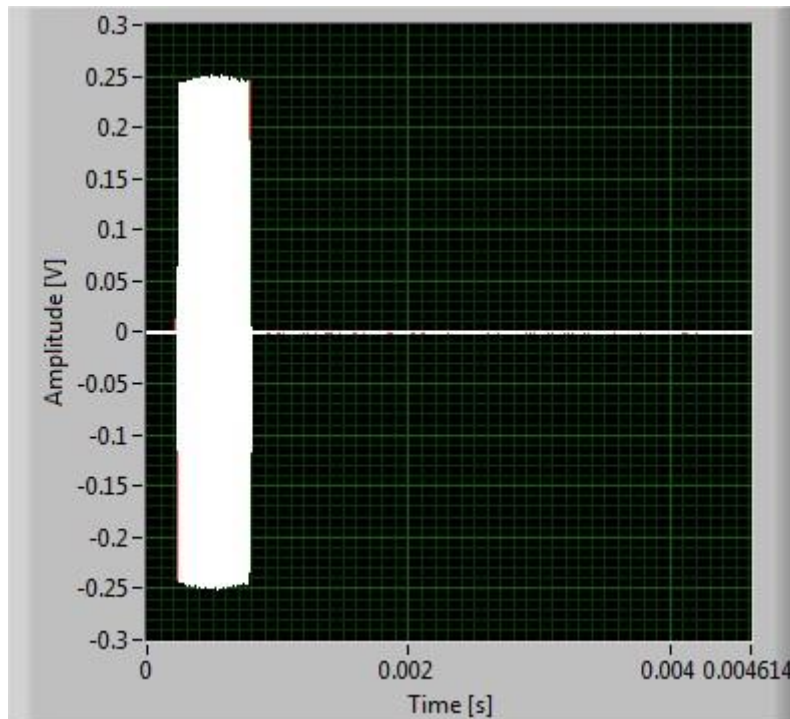


Figura 6.12: Recepción de trama desplazada en tiempo.

Para solucionar el problema, se ha aplicado el método de detección del FCCH periódicamente, logrando así una sincronización constante del sistema. El problema de realizar ese proceso de manera repetida es que aumenta la carga de procesamiento del sistema, por lo que se han realizado modificaciones al método para disminuir la carga, dependiendo del momento de realizar la sincronización:

- Tras un número determinado de tramas
- Tras una nueva adquisición de muestras

#### **6.4.1. Sincronización tras un número determinado de tramas**

Mientras que la primera vez que se realiza la sincronización en tiempo se analizan nueve slots, esta vez se analiza un número menor de slots aprovechándonos de que la señal no se desplaza excesivamente, como se ve en las figuras anteriores. El analizar menos slots conlleva que la carga de procesamiento no sea tan elevada.

Después de analizar un número bajo de tramas, entre 10 y 50, la señal recibida se adelanta o atrasa en tiempo, pero siempre con un desplazamiento menor a la

duración de un slot. Por tanto, cuando realizamos la sincronización en tiempo periódica tan solo tenemos que buscar el FCCH en el slot en el que debería estar y en los slots anterior y posterior.

Por lo tanto se adquieren las muestras equivalentes a tres slots de tiempo y se calcula la correlación entre el flujo de datos recibido y el FCCH de referencia, siguiendo el método usado en la primera sincronización y ejecutando el VI *Correlacion para receptor nuevo optimizado.vi*, desarrollado por Borja Genovés. Una vez detectado el inicio del FCCH, se adquieren y desechan las muestras necesarias para que la siguiente lectura de datos comience en instante adecuado.

#### **6.4.2. Sincronización tras realizar una nueva recepción de datos**

Como se detalla en el apartado 6.3 de este capítulo, el tamaño limitado del buffer obliga a realizar una recepción de datos controlada. Cada vez que el buffer se vacía y se reciben nuevos datos, es necesario sincronizar el receptor de nuevo con la señal recibida.

En este caso el método empleado es exactamente igual al realizado en el módulo *Sync Tiempo.vi*. Debido a que se realiza una recepción de datos completamente nueva, hay que analizar nueve slots de tiempo para detectar un FCCH, por tanto no se puede reducir el tiempo de procesamiento.

### **6.5. Procesamiento de datos**

Cuando el receptor esta sincronizado en tiempo y con el error de frecuencia corregido, está capacitado para analizar las muestras recibidas usando la función de las librerías *Fetch Rx Data (Poly).vi*. El programa ha sido diseñado para que la adquisición de muestras del buffer de recepción se realice de manera repetitiva usando un bucle. En cada iteración del bucle se analiza una trama GSM, correspondiente a ocho slots de tiempo. Previamente se ha de calcular el número de muestras al que equivale una trama.

#### **6.5.1. Eliminación guarda**

Tras adquirir las muestras de la señal, hay que eliminar el periodo de guarda de cada slot para poder demodular posteriormente los datos. En primer lugar se descompone la señal recibida para trabajar directamente con el vector de datos que contiene las muestras.

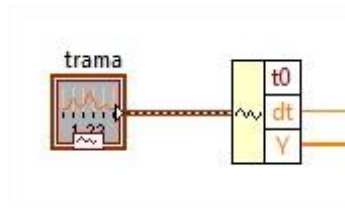


Figura 6.14: Descomposición de la señal adquirida por el USRP.

Para eliminar la guarda se separan los ocho slots de la trama recibida y en cada slot se desechan las muestras correspondientes al periodo de guarda. El método usado es crear un vector por cada slot, recogiendo las muestras de los 148 primeros bits y desechando el resto. Multiplicando los 148 bits por la tasa de muestras por bit empleada se hallan las muestras a adquirir.

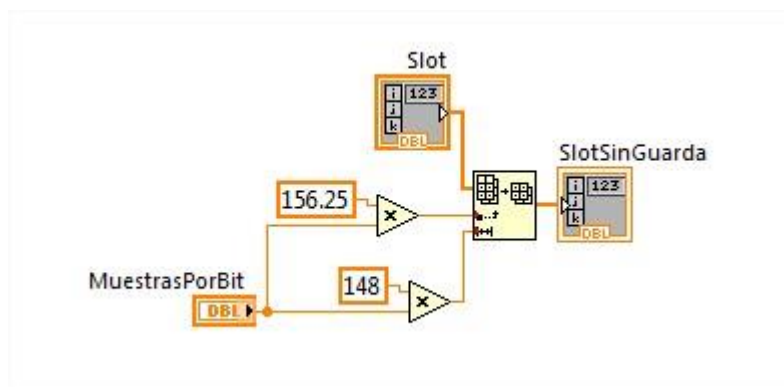


Figura 6.15: Creación de un vector con un slot sin periodo de guarda.

El modulo implementado para realizar este apartado se llama *Eliminar Guarda.vi*. Esta función recibe como parámetros de entrada los datos adquiridos correspondientes a una trama y el número de muestras por bit. El modulo devuelve la señal reconstruida sin los periodos de guarda.

Para desechar los periodos de guarda de los ocho slots de la trama, se extraen ocho sub vectores del vector de la trama indicando la posición de cada uno. Posteriormente los ocho sub vectores se concatenan en un único vector que es añadido a un *cluster* junto a la variable *dt*, que representa el salto en tiempo entre muestras. El diagrama de bloques de LabVIEW de este módulo se muestra en la figura 6.16.

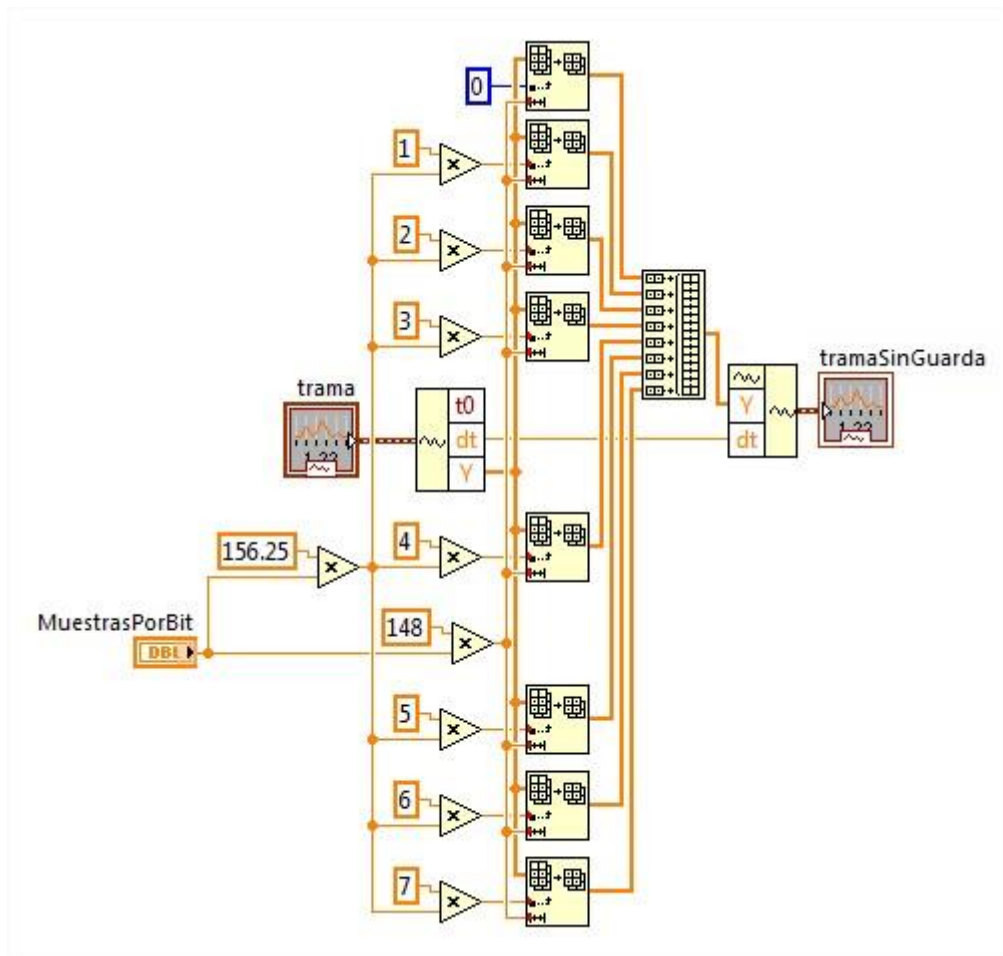


Figura 6.16: Diagrama de bloques del módulo *Eliminar Guarda.vi*

### 6.5.2. Demodulación

Una vez eliminados los periodos de guarda de la trama, el sistema está preparado para demodular la señal recibida. Para ello se ha creado el VI *Demodulador.vi*, siguiendo el ejemplo del programa VI *MT ni5660 MSK Demod (burst detection).vi* incluido en los ejemplos proporcionados por LabVIEW.

El VI creado tiene los siguientes parámetros de entrada:

- *Trama*: señal correspondiente a la trama adquirida.
- *Muestras por símbolo*.
- *Codificación diferencial activa*: variable booleana que indica si está activa o no.
- *Slot*: Intervalo de tiempo a demodular.
- *Reset*: Para reiniciar el demodulador.

El primer paso es descomponer la trama recibida y crear un vector con los datos del intervalo de tiempo que se desea demodular, como se ve en la figura 6.17.

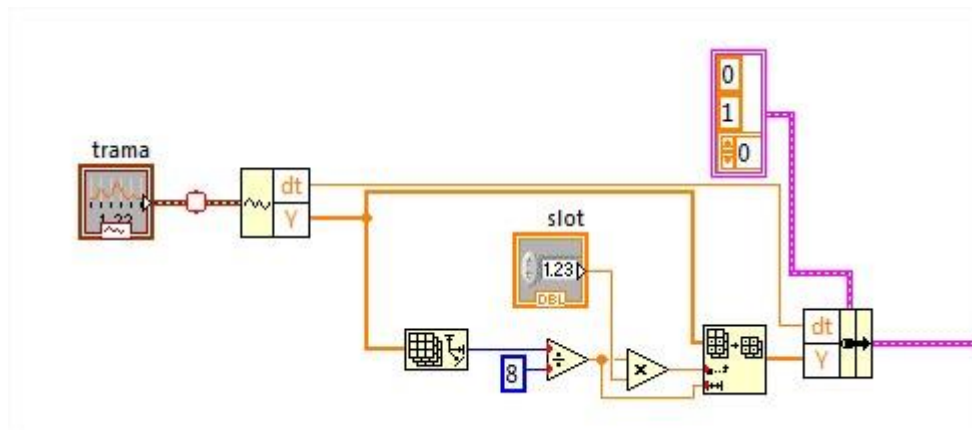


Figura 6.17: Selección del slot de tiempo a demodular.

Para ello se descompone la señal recibida y se trabaja con el vector de muestras. Se calcula el tamaño de dicho vector y se divide entre ocho para dar lugar al número de muestras de un time slot. Los datos del slot deseado son adquiridos del vector de la trama usando la función *Array Subset.vi*. El tamaño del sub vector viene dado por el número de muestras calculado y la posición de este slot en la trama se calcula multiplicando el número de muestras del slot por la variable de control *slot*. Finalmente se contruye un *cluster* con el sub vector adquirido y la componente *dt* de la señal original.

Paralelamente se deben generar los coeficientes del filtro del demodulador y los parámetros del sistema. Esto se realiza con dos funciones de las librerías: *Generate System Parameters.vi* y *Generate Filter Coefficients.vi*. La figura 6.18 muestra los parámetros de entrada de ambas funciones.

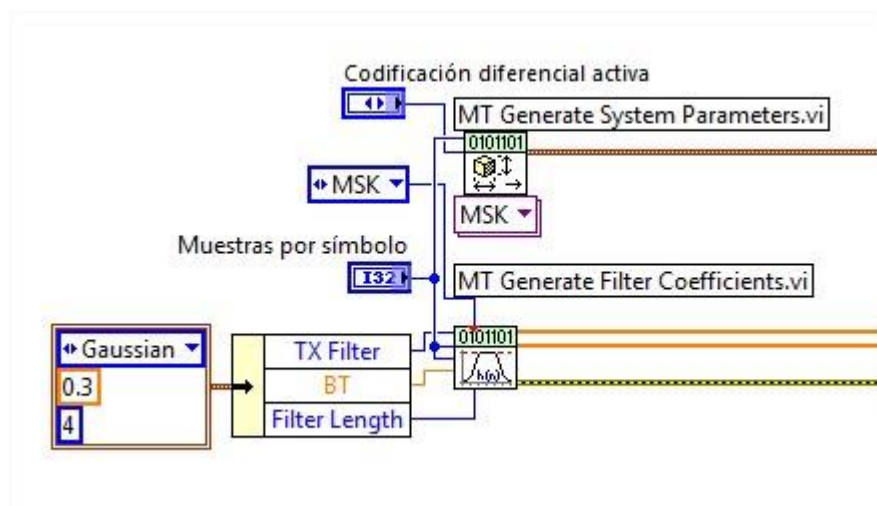


Figura 6.18: Configuración de los parámetros del sistema demodulador y del filtro Gaussiano.

Se observa que el filtro se configura con los siguientes datos:

- Tipo de filtro: *Gaussiano*.



- BT: es el producto del ancho de banda de la señal a -3 dB por el periodo de símbolo. Es 0,3 de acuerdo a las especificaciones de GSM.
- Longitud del filtro.
- Muestras por símbolo: 4, por defecto en el receptor implementado.
- Modulación empleada: MSK.

Los parámetros del sistema son configurados añadiendo como entradas:

- *Codificación diferencial*: activada o no.
- *Muestras por símbolo*: 4.

A continuación todos los datos generados son introducidos al sub VI creado *Demodulate Modificado.vi*. Este VI es una modificación de la función *Demodulate.vi*, incluida en las librerías de LabVIEW. Las modificaciones realizadas en el VI original son las siguientes:

- Codificación diferencial: La conversión de símbolos a bits es realizada en sub VI llamado *mod\_MSK Demod Symbol Demap.vi* (figura 6.19 izquierda). Esta función presenta un modo distinto de realizar la decodificación diferencial respecto a GSM. Para corregirlo se ha reemplazado el VI original por el VI *mod\_MSK Demod Symbol Demap Borja.vi* (figura 6.19 derecha), desarrollado por el cotutor Borja Genovés, con los cambios a la hora de realizar la decodificación diferencial implementados. El nuevo sub VI da la opción además de invertir los bits mediante un parámetro de entrada.

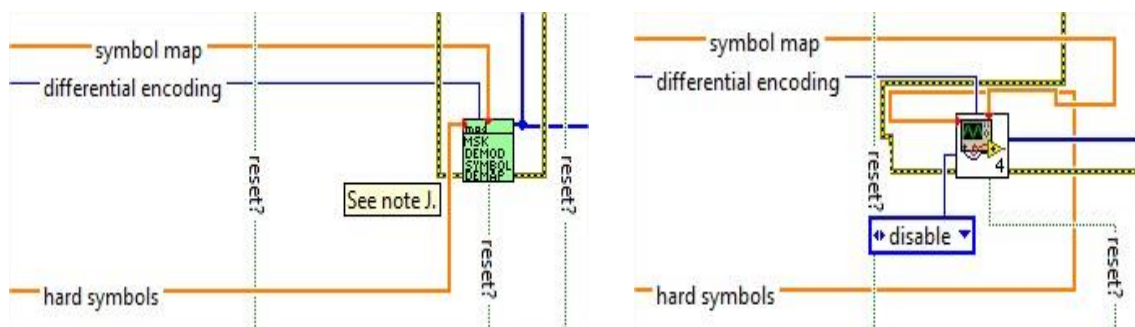


Figura 6.19: Función *Demod Symbol Demap.vi* original y editada.

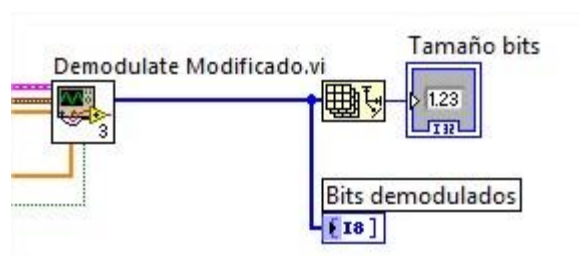


Figura 6.20: Salidas de la función *Demodulador.vi*



Finalmente el VI *Demodulate Modificado.vi* devuelve como salida el flujo de bits demodulados y además se calcula el tamaño del flujo. Las salidas de la función *Demodulador.vi* son las siguientes:

- *Bits demodulados*: que contiene el flujo de bits demodulados.
- *Tamaño bits*: referencia el número de bits del slot demodulado.
- *Error*: que devuelve la información sobre los posibles errores a la hora de demodular.

### 6.5.3. Detección del patrón

Tras realizar la demodulación de los bits del slot deseado, el flujo de bits demodulados es devuelto al programa principal donde se realiza la detección del patrón deseado. El vector que contiene los bits se convierte a un string o cadena de caracteres mediante el VI *Array to Spreadsheet String*, proporcionado por LabVIEW en sus librerías.

Después se realiza la búsqueda del patrón en la cadena de bits mediante la función de LabVIEW *Match Pattern*. Esta función busca el patrón indicado en el flujo de bits demodulados y devuelve el patrón encontrado.

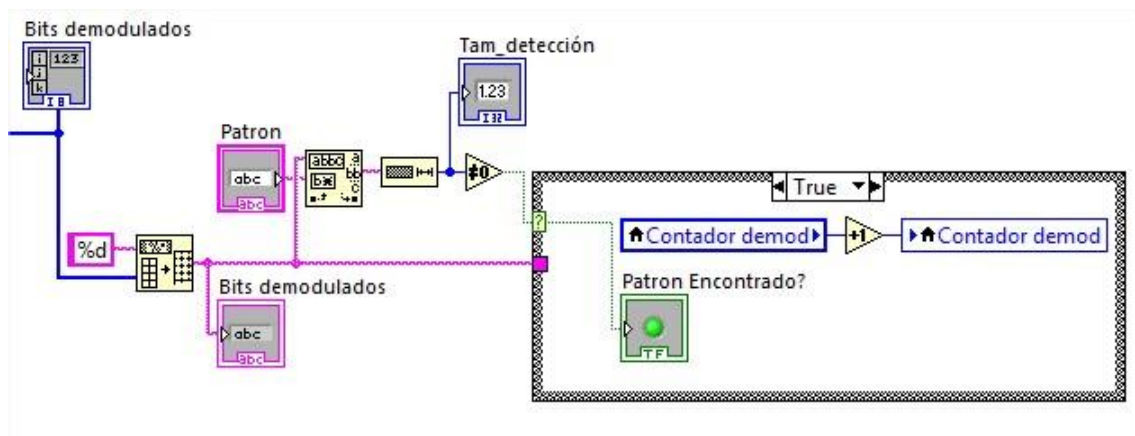


Figura 6.21: Diagrama de bloques del detector de patrones.

Finalmente si el tamaño del patrón encontrado no es nulo se ejecuta una estructura que actualiza el contador de demodulaciones realizadas y cambia a *True* el valor de la variable local *Patron Encontrado?*.

Si el patrón no es detectado la iteración del bucle finaliza y se pasa a la siguiente en la que se analizará una nueva trama o se realizara una sincronización en tiempo.

## 6.6. Transmisor

En consecuencia a las consideraciones previas descritas al inicio de este capítulo, el transmisor ha sido diseñado de modo que funcione paralelamente a la ejecución del receptor y que transmita de manera constante un canal FCCH en el primer intervalo de tiempo de cada trama. Una vez que en el receptor se detecte el patrón buscado, se activará la transmisión del intervalo de tiempo TN3, en el que se transmitirá un canal TCH.

Para la implementación del transmisor en el programa realizado, se ha hecho una adaptación del VI *BTS-Control Básico.vi* desarrollado por el cotutor Borja Genovés. La adaptación consiste en gestionar todas las variables de control necesarias desde la interfaz principal del programa. Estas variables son introducidas como parámetros de entrada en el VI original; son las siguientes:

- *IP Tx*: es la dirección IP del USRP asociado a la transmisión.
- *Antenna Tx*: la antena del USRP que deseamos usar. Siempre será TX1.
- *freq Tx*: es la frecuencia a la que trabajará el transmisor. En Herzios.
- *gain Tx*: es la ganancia en dB de la transmisión.
- *Patrón TCH*: Secuencia de entrenamiento de los canales TCH.
- *Slot*: del 0 al 7. Referencian el canal que se desea transmitir.

Los slots a transmitir pueden ser seleccionados por el usuario mediante la interfaz de transmisión en el menú principal, pero en este trabajo la configuración elegida ha sido la mostrada en la figura 6.22.

Hay que destacar que la presencia del canal AGCH es meramente ilustrativa debido a que no se ha realizado su implementación. En su lugar el canal que se transmite en el slot 3 es un TCH. La configuración mostrada solo se activa una vez que el receptor detecta el patrón buscado. Mientras tanto, el único canal que se transmite es el FCCH en el slot 0.

The image shows a software window for configuring the transmitter (Tx). It has a tabbed interface with 'Rx' and 'Tx' tabs, where 'Tx' is active. The configuration parameters are as follows:

- IP Tx:** 192.168.10.6
- freq Tx:** 550M
- Antenna Tx:** TX1
- gain Tx:** 10
- Patrón TCH (del 0 al 15):** 0
- Slot 0:** FCCH
- Slot 1:** DISABLED
- Slot 2:** DISABLED
- Slot 3:** AGCH
- Slot 4:** DISABLED
- Slot 5:** DISABLED
- Slot 6:** DISABLED
- Slot 7:** DISABLED

Figura 6.22: Configuración del transmisor.

### Implementación

Cuando se inicia la ejecución del programa, el receptor y el transmisor se activan en el mismo instante de tiempo. El transmisor es inicializado con los parámetros configurados desde la interfaz de usuario y acto seguido se inicia la transmisión de un canal FCCH en el intervalo de tiempo  $TN_0$ .

La transmisión se mantiene sin cambios hasta que el receptor detecta el patrón deseado en la señal recibida. Entonces se activa la transmisión del resto de slots tal como se hayan configurado en la interfaz; en el caso de este TFG se transmite un canal TCH en el slot 3.

Este proceso es gestionado mediante una estructura de casos cuyo parámetro de control es la variable local *Patrón Encontrado?*. Cuando el valor de esta variable es falso todos los slots permanecen inactivos, excepto el slot 0 que se encuentra transmitiendo el FCCH. Si la variable tiene valor verdadero los slots transmiten el canal que corresponda a la configuración realizada en la interfaz.

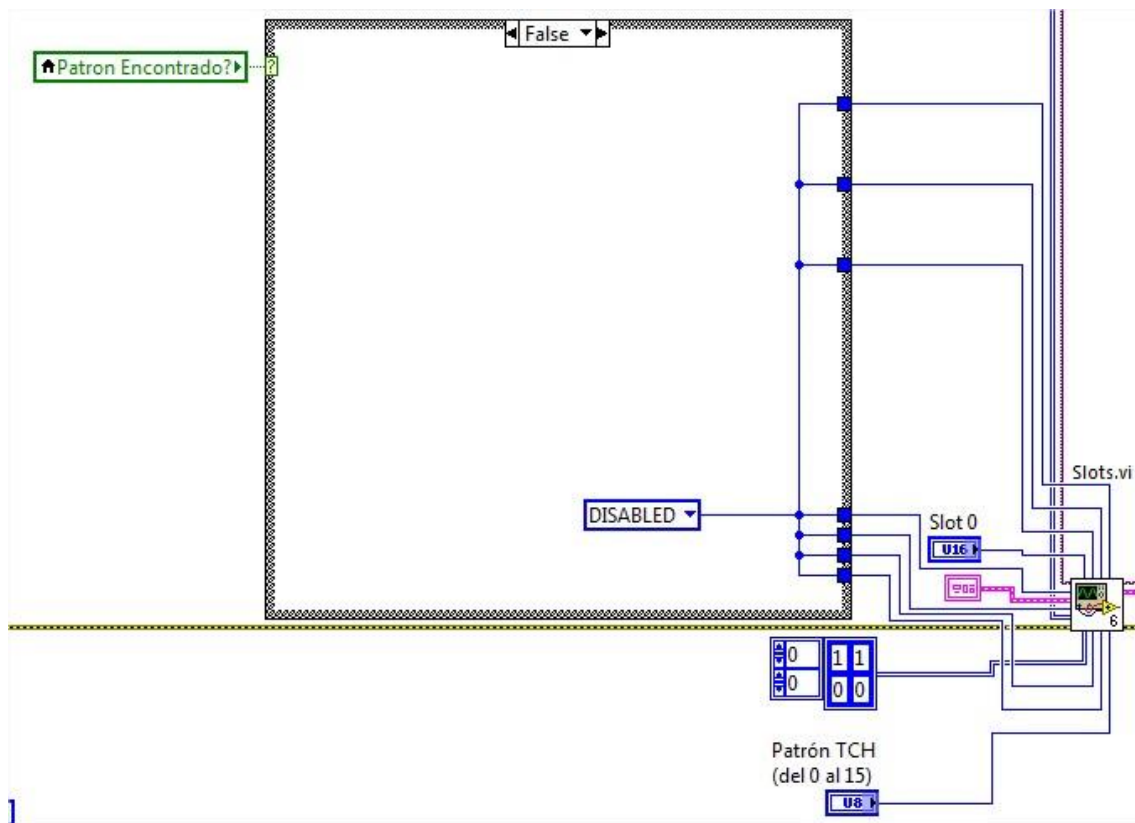


Figura 6.23: Diagrama de bloques del transmisor desactivado.

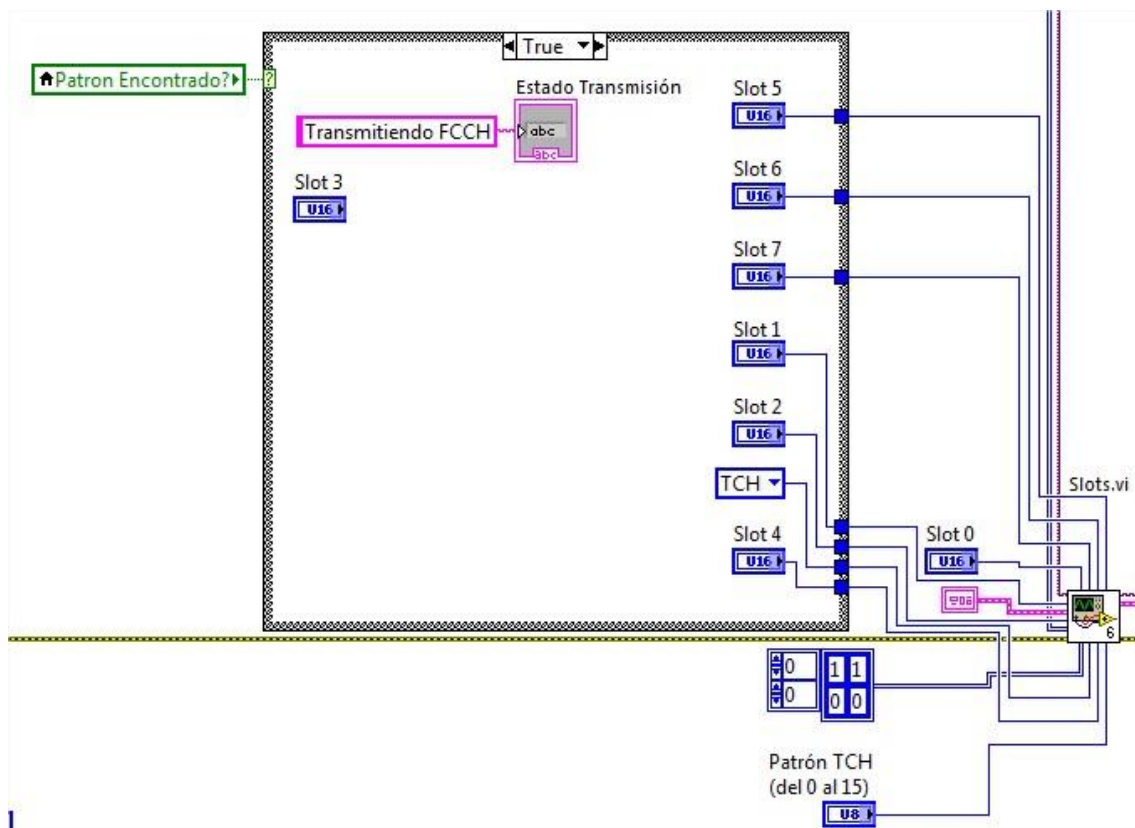


Figura 6.24: Diagrama de bloques de la respuesta del transmisor.

## 7. Pruebas y resultados

En este capítulo se muestran los resultados y pruebas de los distintos módulos implementados en este trabajo y descritos en el capítulo anterior. Los resultados analizados son los siguientes:

- Sincronización en tiempo: tanto al inicio de la comunicación como en el resto de la ejecución del programa.
- Ajuste de frecuencia
- Eliminación del periodo de guarda
- Demodulación
- Detección de patrón
- Respuesta del transmisor

### 7.1. Sincronización en tiempo

Este apartado muestra los resultados obtenidos tras realizar la detección de un FCCH para sincronizar el receptor en tiempo. Como se detalla en el anterior capítulo, esta sincronización en tiempo se realiza en tres momentos distintos:

- Al iniciar el receptor.
- Periódicamente, tras un número determinado de tramas.
- Tras realizar una nueva adquisición de muestras.

Por tanto, el receptor debe estar sincronizado en todo momento, detectando el canal lógico FCCH en el primer slot de tiempo de la trama adquirida. Si no se realizará la sincronización en tiempo, la adquisición de datos tendría la forma de la figura 7.1, en la que se puede observar que la trama recibida no está sincronizada, dado que uno de los slots no se adquiere completamente.

Sin embargo, la figura 7.2 muestra los dos slots en el instante de tiempo adecuado tras aplicar el método descrito anteriormente para sincronizar el sistema, siendo el primero de los slots un FCCH.

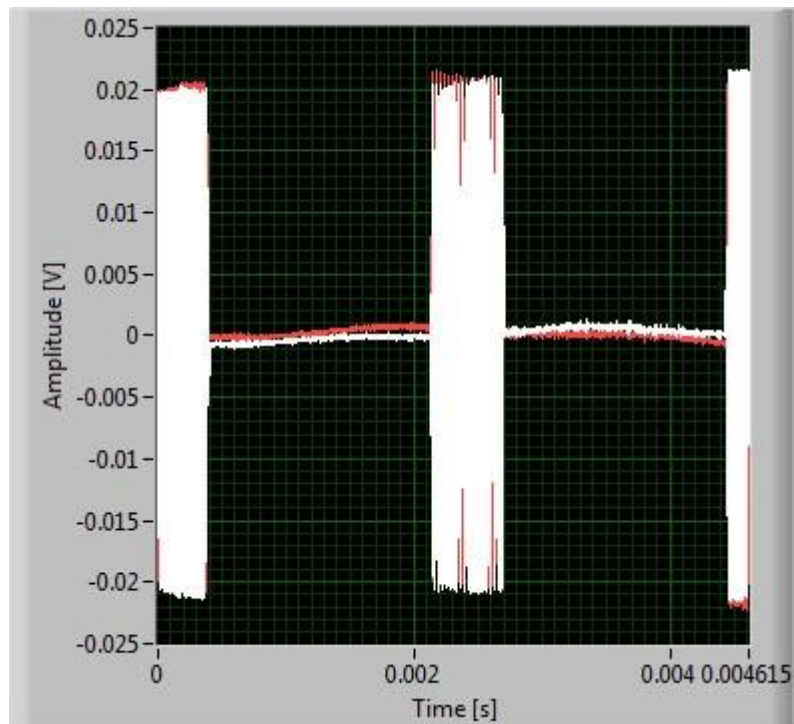


Figura 7.1: Trama recibida no sincronizada en tiempo.

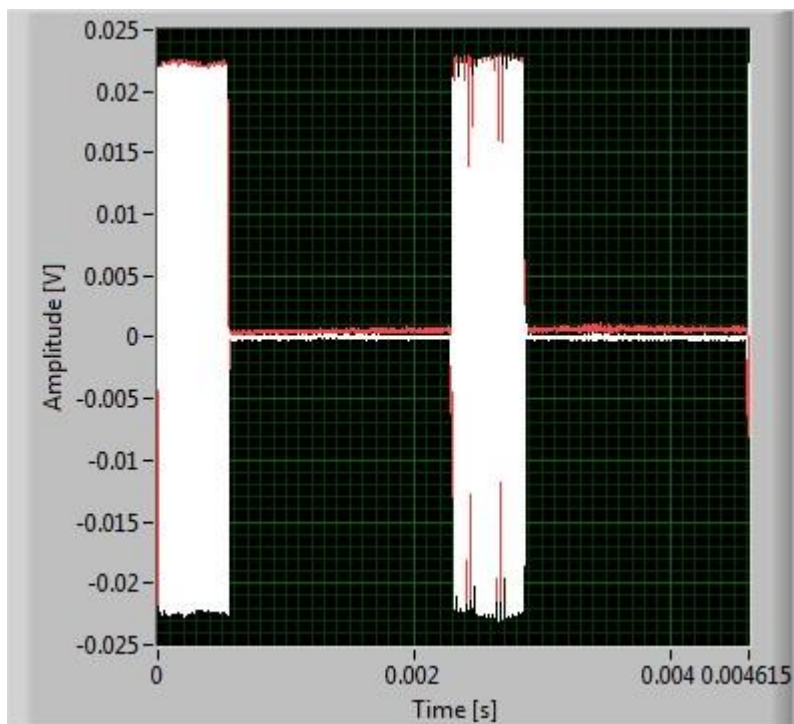


Figura 7.2: Trama recibida sincronizada en tiempo.

A continuación se van a mostrar los gráficos mostrados en la interfaz de usuario en diversos momentos de la ejecución y tras generar distintas señales tanto en el generador de señales como usando un USRP y la implementación *BTS-Control Basico.vi*, desarrollado por Borja Genovés.



### 7.1.1. Recepción de dos slots enviados por el generador de señales

Desde el generador de señales vectoriales se configura una trama GSM de la manera que ilustra la figura 7.3. En el primer slot de tiempo se introduce una ráfaga de corrección de frecuencia, que incluye un FCCH. El segundo slot transmite una ráfaga normal. En la figura 7.4 se observa como la señal es recibida correctamente en tiempo, después de realizar la sincronización en tiempo.

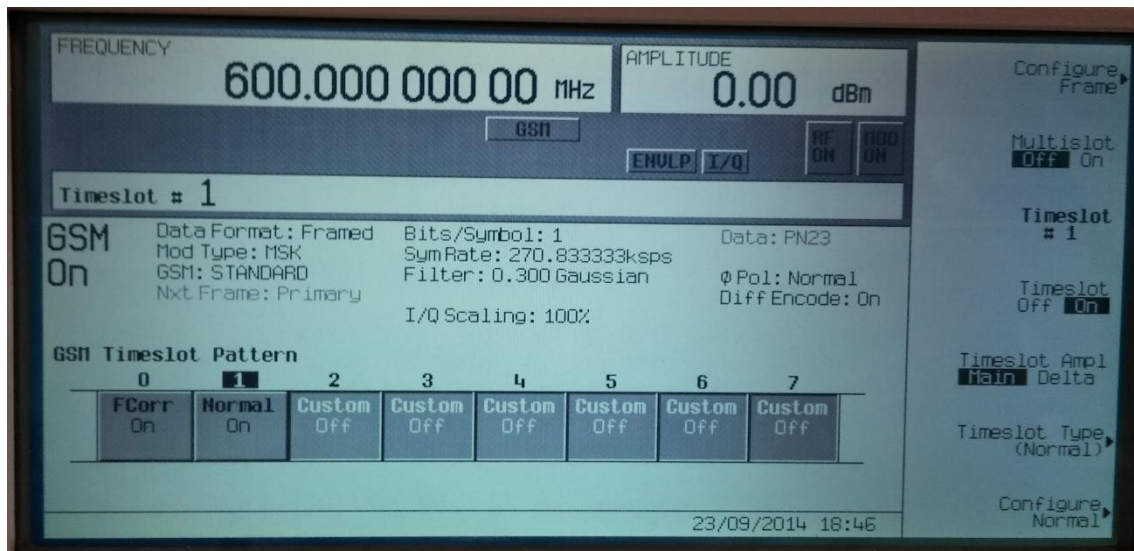


Figura 7.3: Trama GSM en el generador de señales con dos slots ocupados.

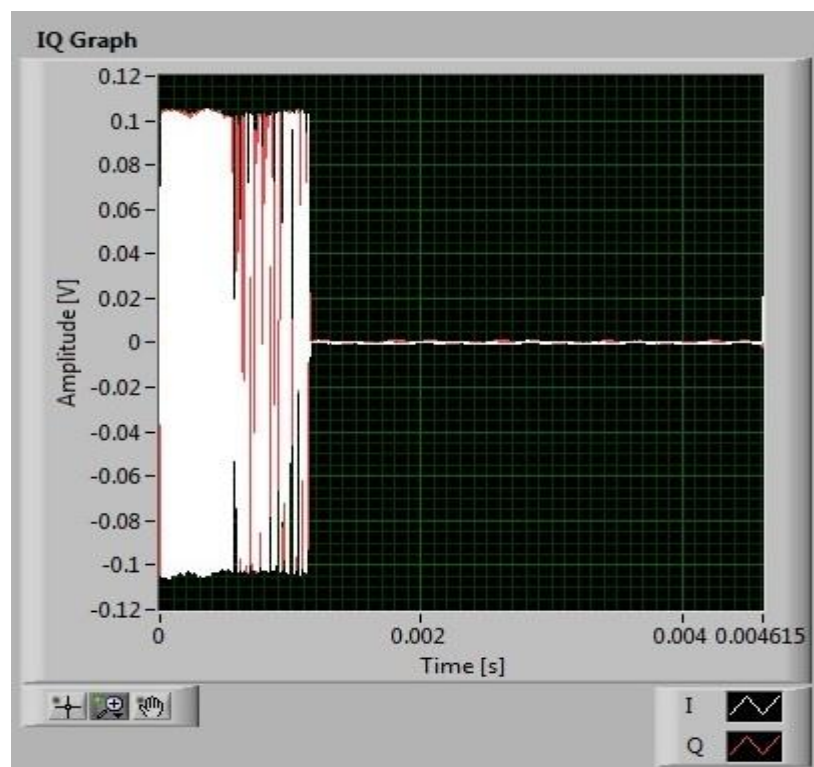


Figura 7.4: Componentes IQ de dos slots sincronizados en tiempo.

### **7.1.2. Recepción de ocho slots enviados por el generador de señales**

Para este apartado se ha configurado en el generador de señales vectoriales una trama con sus ocho slots ocupados, como se muestra en la figura 7.5. El slot TN0 porta una ráfaga FB, el slot TN7 es una trama configurada manualmente que emite el siguiente patrón de bits: 1100, y el resto de los slots transmiten ráfagas DB.

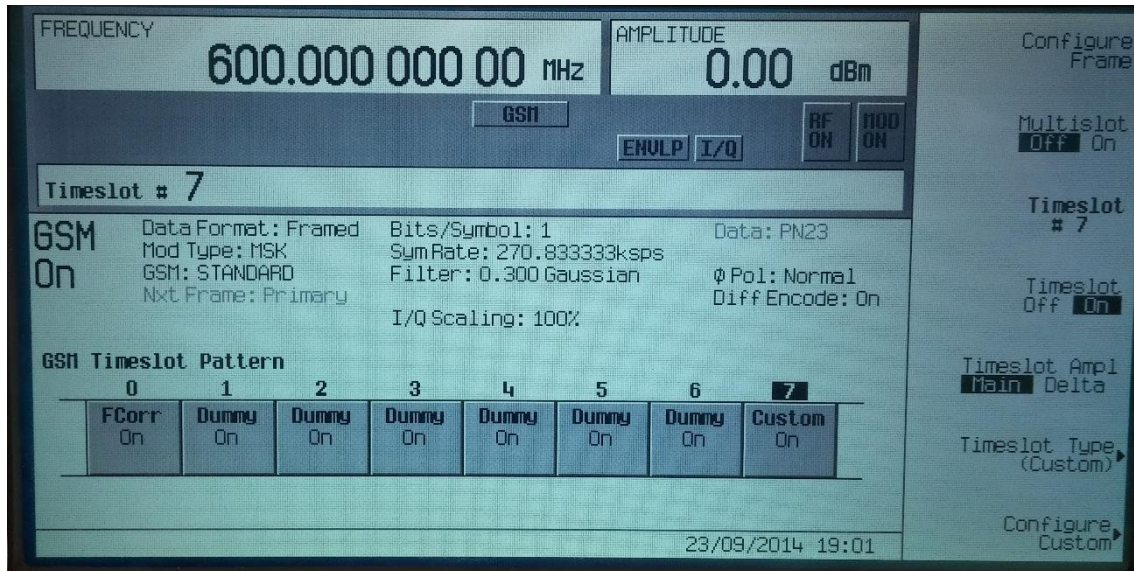


Figura 7.5: Configuración trama GSM en el generador de señales con ocho slots ocupados.

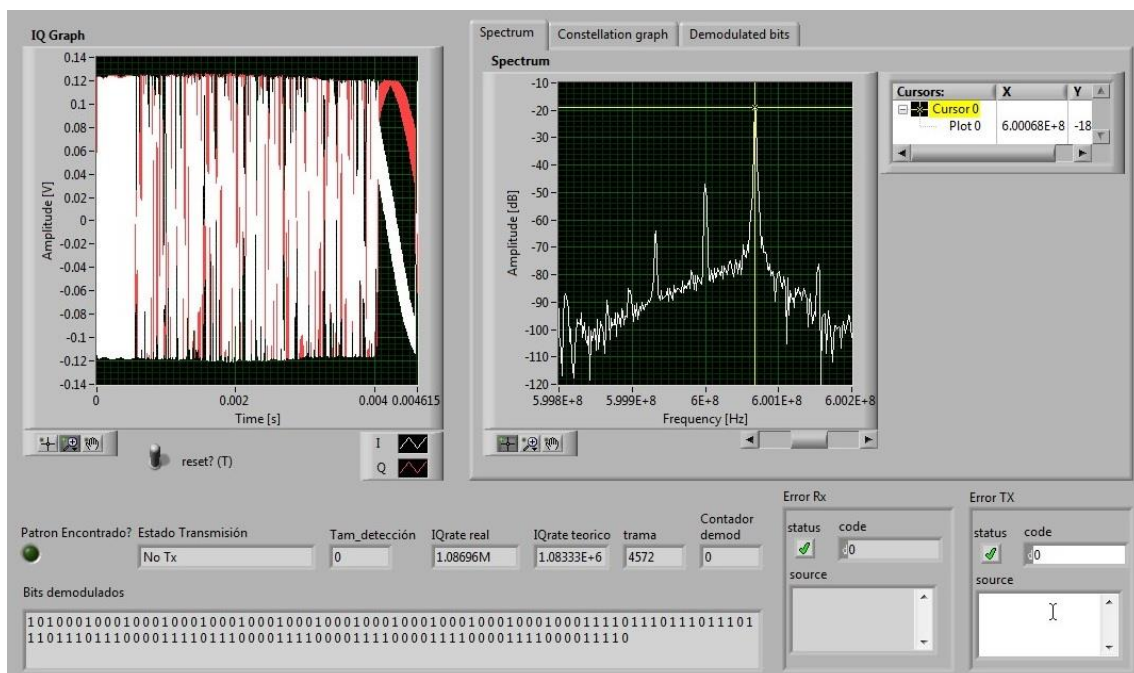


Figura 7.6: Resultados de la recepción de ocho slots.



El resultado obtenido en un instante de tiempo aleatorio de la ejecución del programa muestra que la adquisición de las muestras recibidas permanece sincronizada en todo momento, como se ve en la figura 7.6. Se observa que la trama analizada es la número 4572, lo que indica que el receptor permanece sincronizado tras un periodo de tiempo en el que se han realizado varias adquisiciones de muestras.

Además se puede comprobar que el primer slot de la trama mostrada es el FCCH ya que el espectro del slot TNO muestra un tono aproximadamente 68 kHz por encima de la frecuencia portadora.

### 7.1.3. Recepción de tres slots enviados por un URSP

Para probar la sincronización de tramas enviadas desde un USRP, el VI *BTS-Control Básico.vi* se ha configurado de la manera que muestra la figura 7.7.

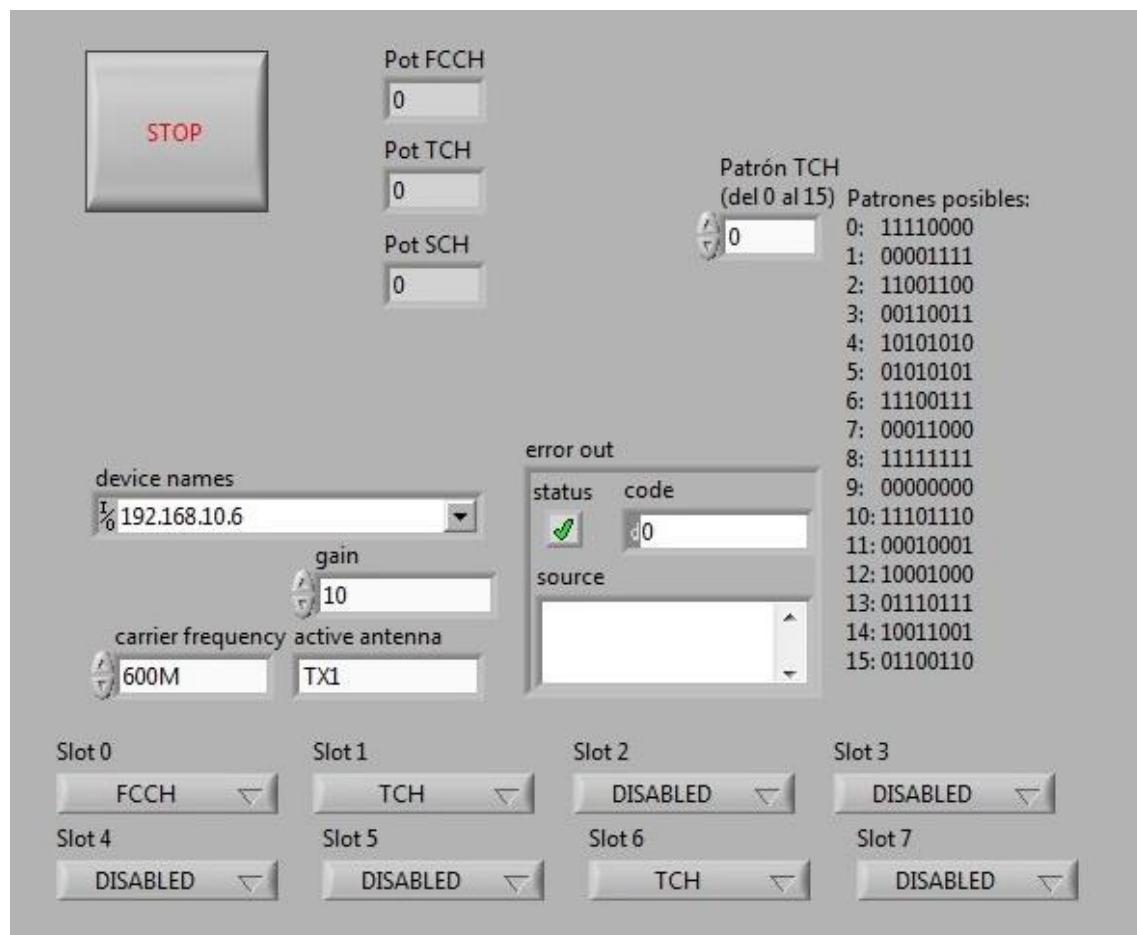


Figura 7.7: Configuración de la trama GSM con un FCCH y dos TCH.

El resultado obtenido en la figura 7.8 muestra la recepción de la señal en un instante avanzado de la ejecución del programa, la trama analizada número 3658. Se ve como el gráfico de la señal en tiempo muestra los slots sincronizados

en tiempo y tal como se han configurado y ademas el espectro del FCCH esta desplazado por encima de la frecuencia portadora como se establece en el estandar.

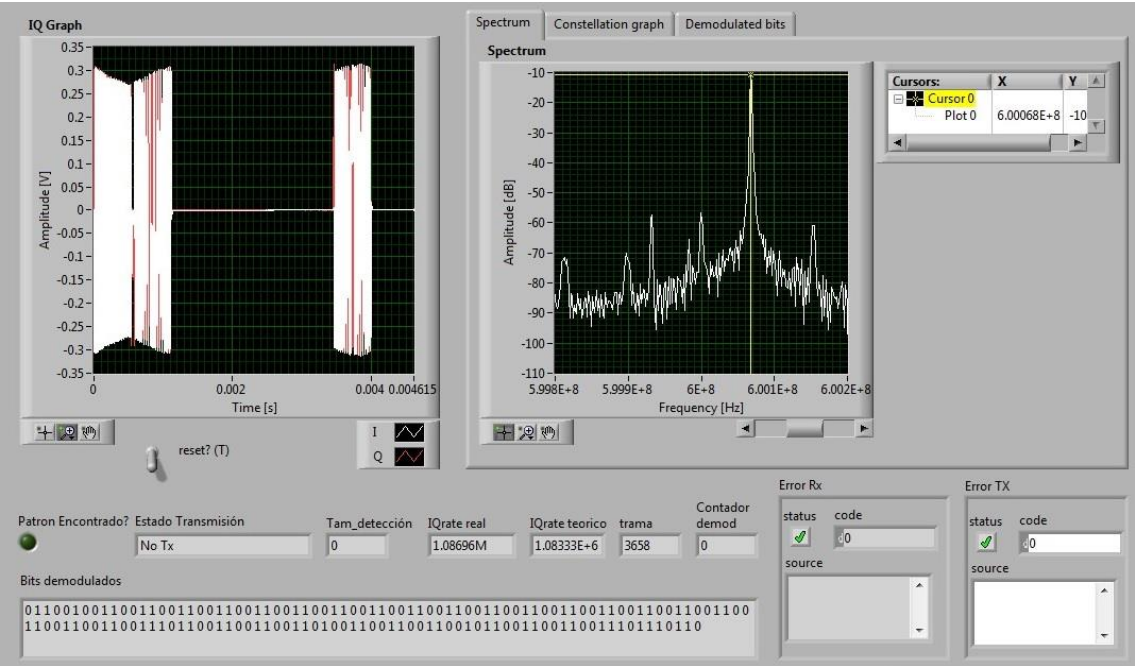


Figura 7.8: Resultados de la recepción de tres slots.

#### 7.1.4. Recepción de cuatro slots enviados por un URSP

Esta prueba demuestra que el receptor es capaz de detectar correctamente el FCCH aunque este se encuentre rodeado de otros slots ocupados. Para ello se han configurado la trama GSM del modo mostrado en la figura 7.9, con un FCCH en el slot TN0 y canales TCH en los slots TN1, TN6 y TN7.



Figura 7.9: Configuración trama con cuatro slots

Los resultados obtenidos en la figura 7.10 muestran que la detección del canal FCCH se realiza correctamente. Los slots se encuentran sincronizados en tiempo y el FCCH muestra su tono desplazado en frecuencia de la manera esperada.

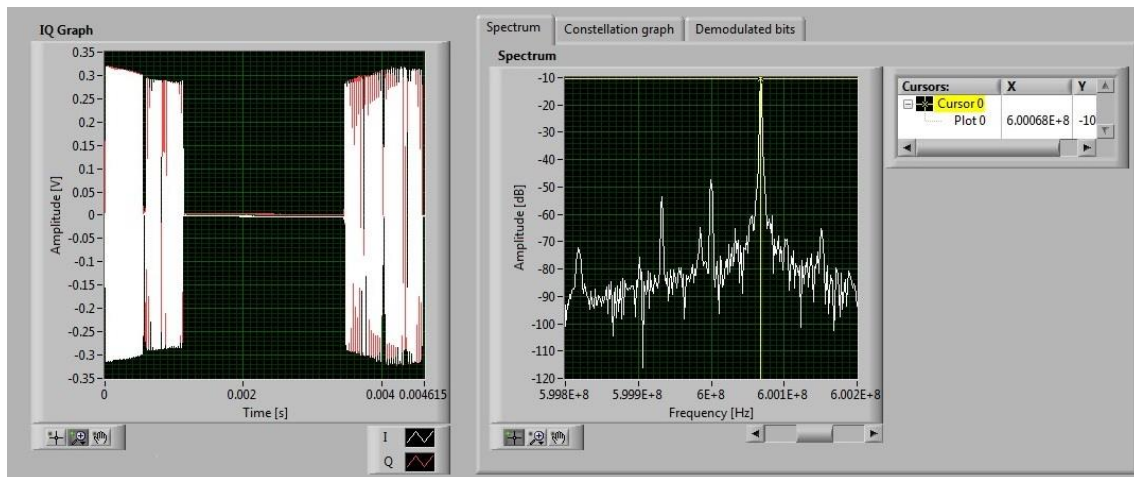


Figura 7.10: Espectro de un FCCH y trama en el dominio del tiempo

### 7.1.5. Recepción tras una nueva adquisición de datos

Cuando el buffer de recepción de muestras se vacía, se realiza una nueva recepción de datos, como se detalla en el apartado 6.4.2 de este trabajo. Esta adquisición de muestras provoca que la sincronización en tiempo del sistema se pierda completamente, por tanto se deben realizar los cálculos necesarios para sincronizar el receptor una vez que reciba las nuevas muestras.

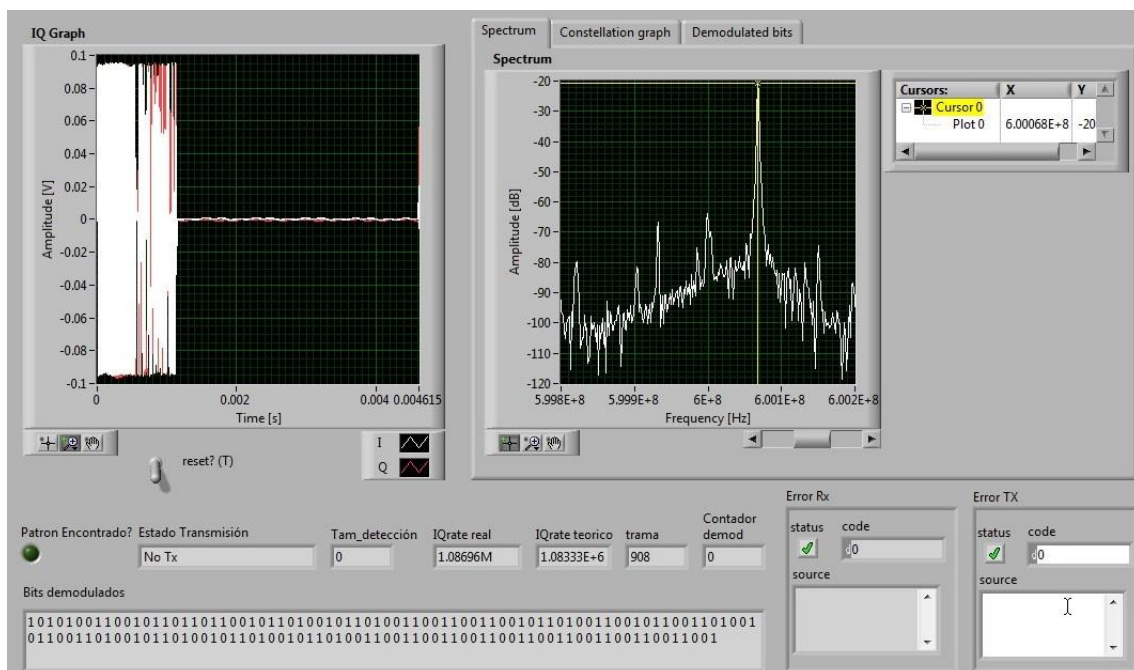


Figura 7.11: Resultados tras nueva adquisición de muestras

La figura 7.11 muestra la primera trama analizada tras realizar una nueva recepción de datos. Esta captura se realizó tras introducir un *breakpoint*

(interrupción) después de la adquisición de nuevas muestras en el VI *Control buffer.vi*. La interrupción se detalla en la figura 7.12.

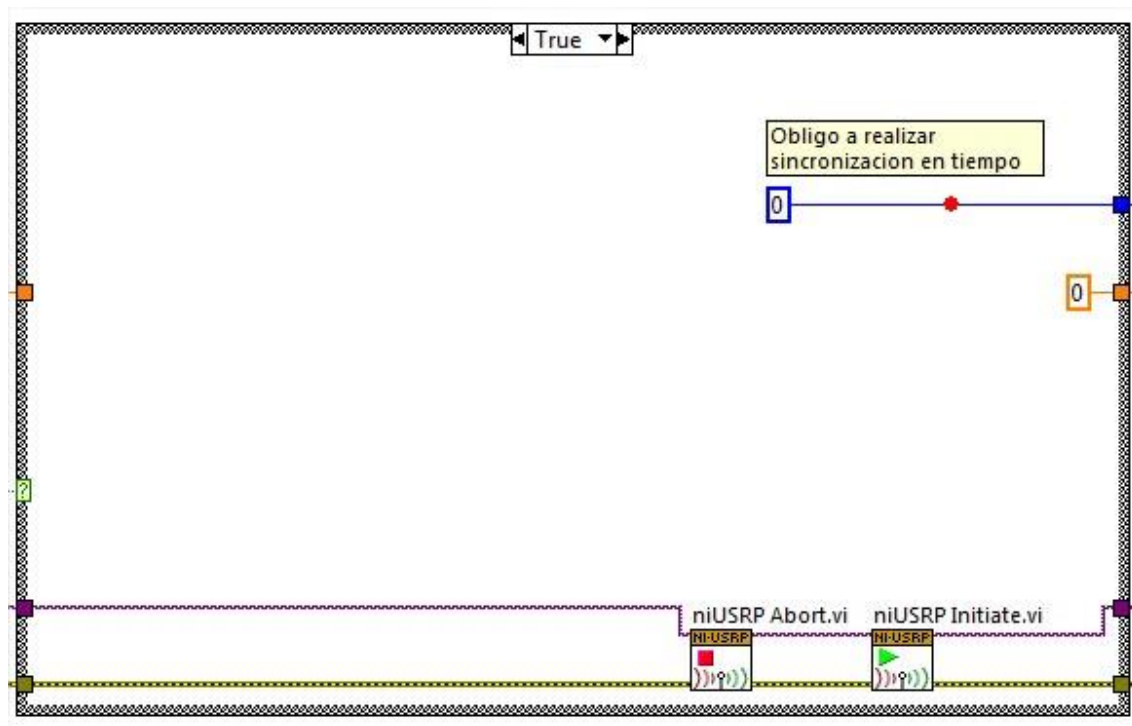


Figura 7.12: *Breakpoint* tras ejecutar *Initiate.vi*

## 7.2. Ajuste de frecuencia

Este apartado muestra el desplazamiento en frecuencia que introducen los equipos usados y su posterior corrección. Para ello se han comprobado los resultados mostrados en sub VI *CorregirOffsetFrec.vi*.

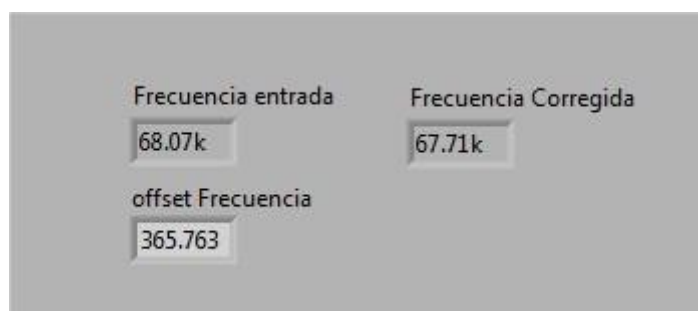


Figura 7.13: Corrección del desplazamiento en frecuencia

La figura 7.13 muestra los valores obtenidos tras analizar la frecuencia del tono de mayor amplitud de un slot con un FCCH. La frecuencia de entrada da un resultado de 68,07 kHz (resultado redondeado), por lo que comparando con los 67,708 kHz que debería dar teóricamente devuelve un desplazamiento en frecuencia de 365 Hz, tal como se muestra en la figura 7.13. El indicador



*Frecuencia Corregida* muestra un valor de 67,71 kHz, que es el resultado redondeado del valor teórico. Por tanto el modulo corrector de frecuencia ajusta la frecuencia de la señal correctamente.

### 7.3. Eliminación del periodo de guarda

Para mostrar el resultado de la supresión del periodo de guarda realizado por el VI *Eliminar Guarda.vi*, se ha hecho uso de una trama enviada por el generador de señales vectoriales transmitiendo dos slots consecutivos. Las componentes IQ de estos dos slots pueden verse en dominio del tiempo en las figura 7.14 y 7.15. Sin embargo la primera de ellas muestra la señal sin procesar, es decir, con el periodo de guarda intacto. La figura 7.15 muestras las componentes IQ de los dos slots sin los periodos de guarda, es decir solo se representan las muestras correspondientes a los 148 bits de cada slots.

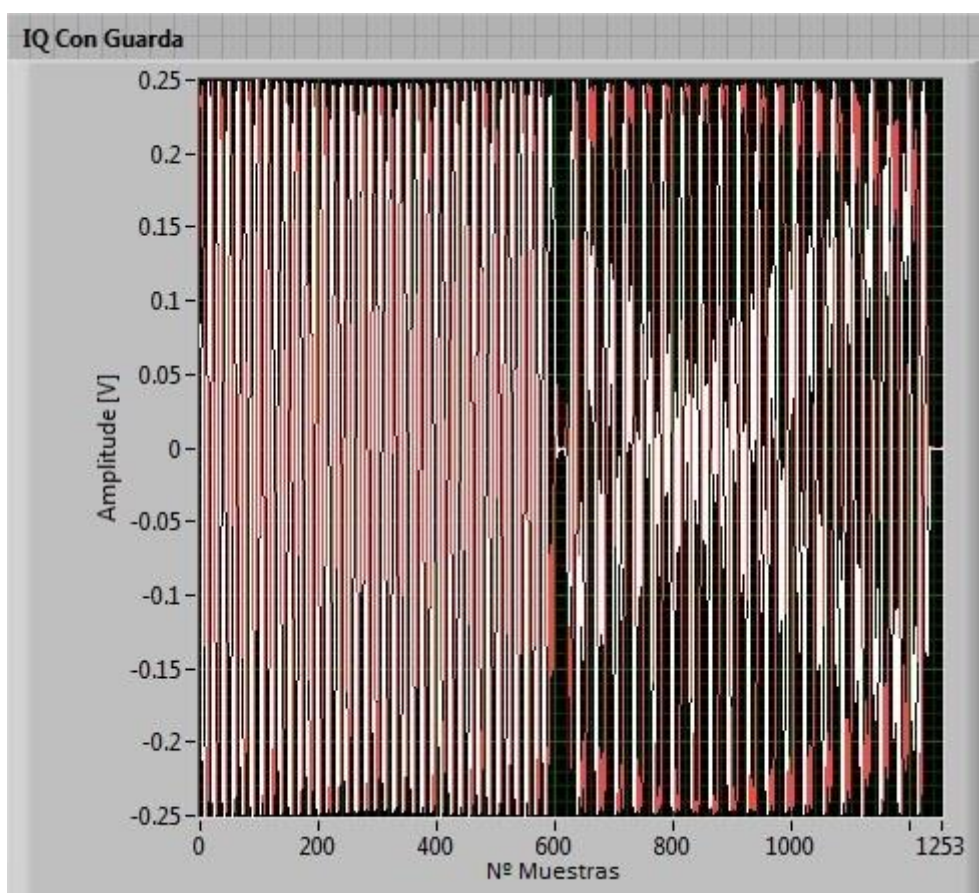


Figura 7.14: Dos slots con los periodos de guarda

Puede verse en la figura 7.14 que el número de muestras mostrado es de 1253, es decir, prácticamente el número de muestras correspondientes a dos slots de tiempo. Dos slots de tiempo con sus periodos de guarda equivalen a 312,5 bits; con una tasa de muestreo de 4 muestras por bits las muestras correspondientes son 1250 muestras.

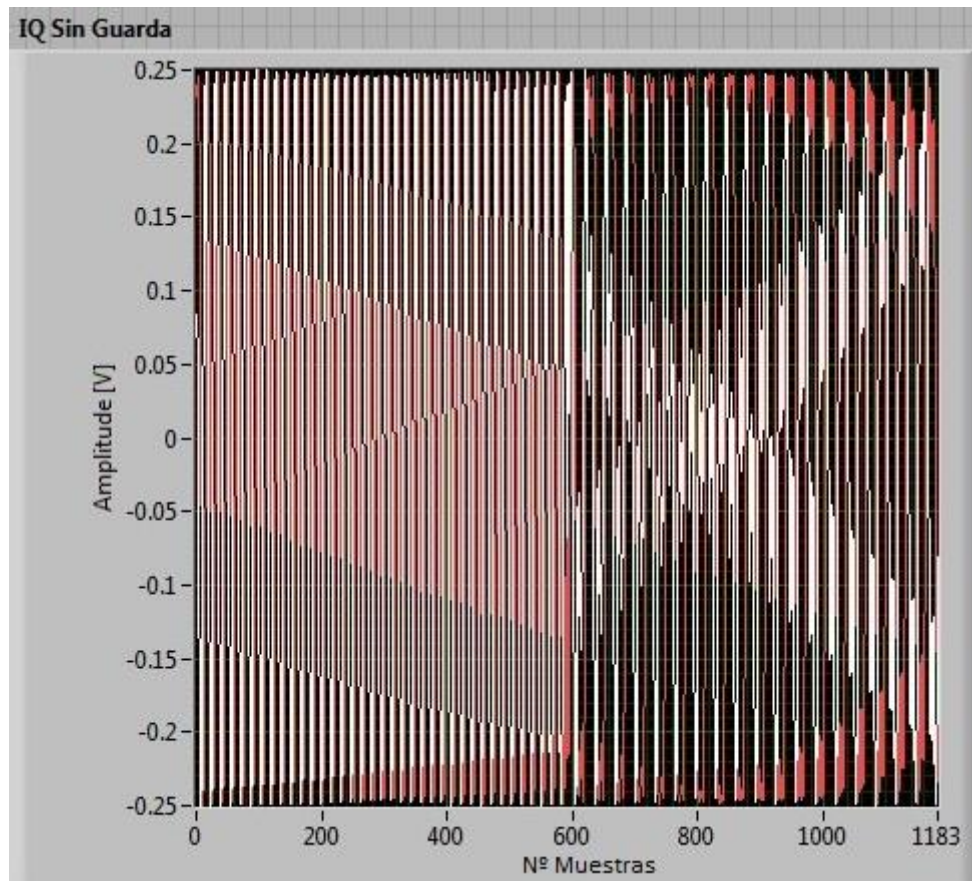


Figura 7.15: Dos slots sin los periodos de guarda

La figura 7.15 muestra las muestras de los dos slots sin los periodos de guarda mostrados en la figura 7.14. Obviamente el número de muestras ha disminuido a 1183, una muestra menos que las muestras correspondientes a los 296 bits que contienen ambos slots.

Estas ligeras variaciones en el número de muestras son debidas a que la tasa de muestreo de los USRP no es exactamente la misma que se introduce. La variable *IQrate real*, mostrada en la interfaz de usuario muestra la variación respecto a la tasa teórica.

## 7.4. Demodulación

Los resultados obtenidos en este apartado corresponden al VI *Demodulador.vi*. Para comprobar el funcionamiento del demodulador se han enviado slots con distintos patrones de bits.

### 7.4.1. Demodulación del patrón 1111 0000

Para probar este apartado se configura el generador de señales para que envíe en el slot TN0 una ráfaga FB y en el slot 4 una trama editada. La trama editada corresponde a 148 bits que siguen el patrón: 0000 1111.

El resultado de la demodulación se muestra en la figura 7.16. Se observa que el flujo de bits muestra el patrón enviado, si bien tiene algunos errores de demodulación, como se muestra al final de la primera línea. Sin embargo, pese a esos errores el resultado en general es bueno y se logra realizar una demodulación del patrón enviado.

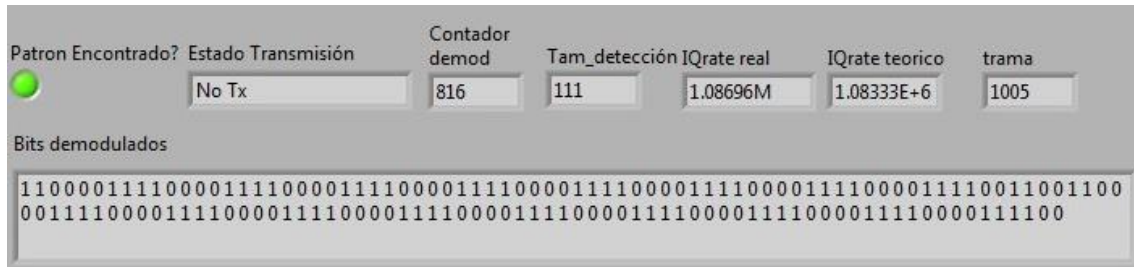


Figura 7.16: Flujo de bits demodulados del patrón 0000 1111

Para mostrar el rendimiento de la demodulación y posterior detección del patrón, se ha realizado la recepción y análisis de unas 1000 tramas GSM. El indicador *Contador demod* muestra que de esas 1005 tramas analizadas, el receptor ha detectado el patrón buscado 816 veces, teniendo el patrón buscado un tamaño de 62 bits.

#### 7.4.2. Demodulación del patrón 11 00

El generador de señales se configura en este caso para que envíe en el slot TN0 una ráfaga FB y en el slot TN5 una trama editada. La trama editada corresponde a 148 bits que siguen el patrón: 00 11 00 11, como se puede ver en la figura 7.17.

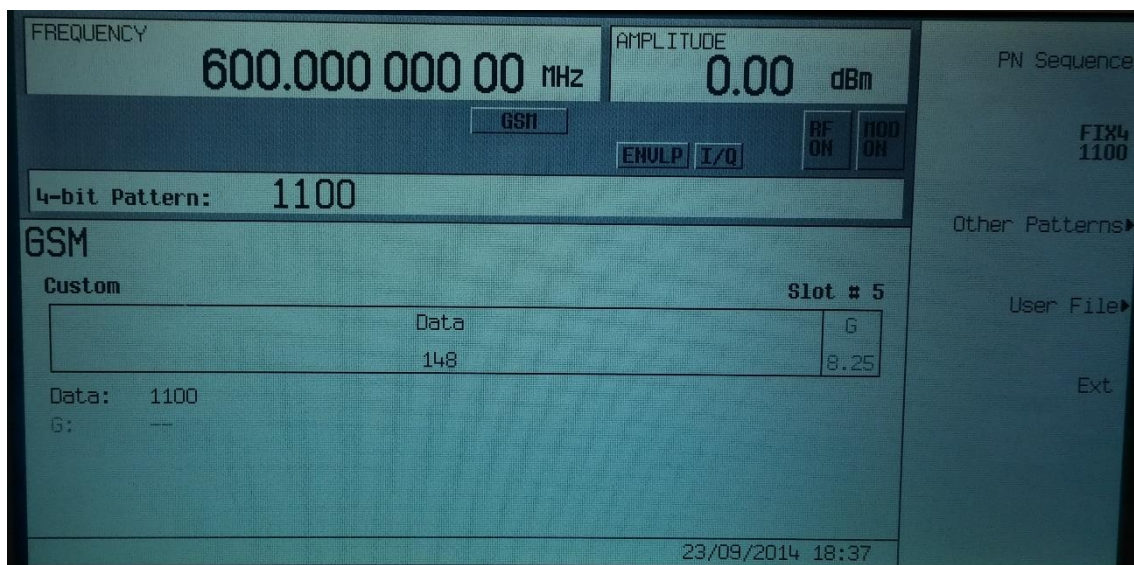


Figura 7.17: Configuración del patrón 1100



En la figura 7.18 se muestra la gráfica que representa las componentes IQ en función del tiempo, demostrando que se recibe la señal tal como es generada.

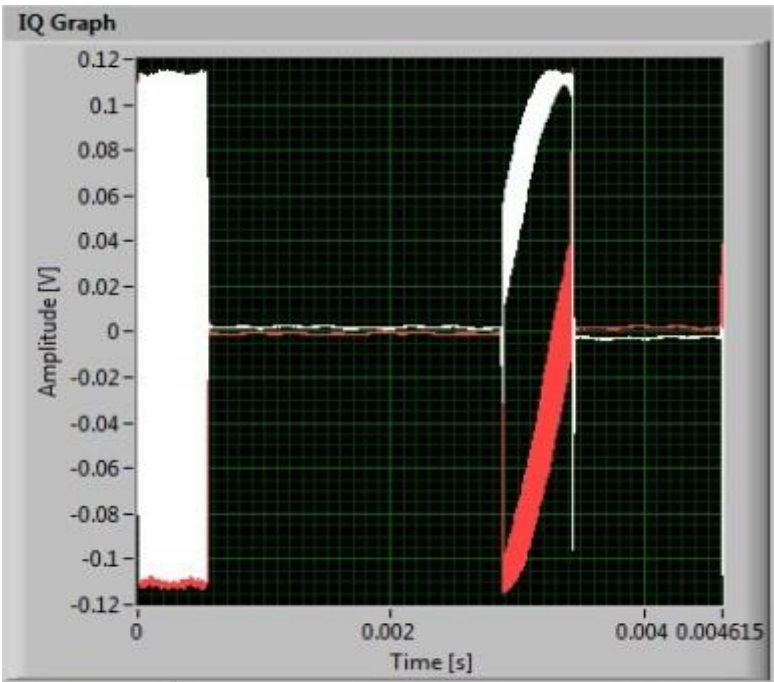


Figura 7.18: Señal en tiempo del patrón 0011 y el FCCH

La figura 7.19 presenta el flujo de bits resultado de la demodulación en el que se comprueba que esta se realiza de forma eficaz, sin apenas errores. Además se muestra el resultado de la detección del patrón buscado (figura 7.20). De 1000 tramas analizadas, se ha detectado el patrón 844 veces (indicador *Contador demod*), siendo el tamaño del patrón de 128 bits como muestra el indicador *Tam\_detección*.

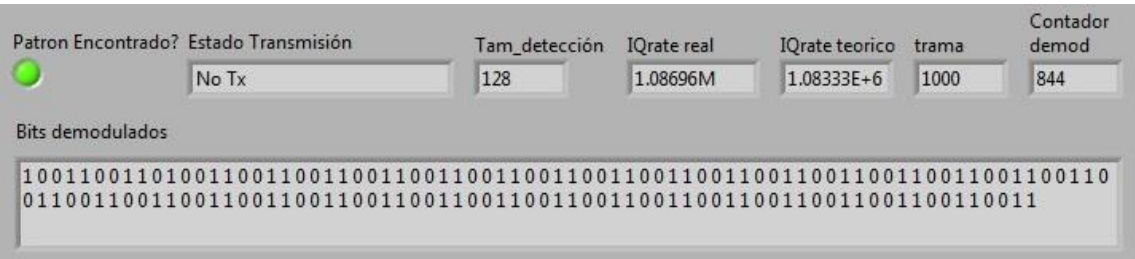


Figura 7.19: Flujo de bits demodulados del patrón 0011.

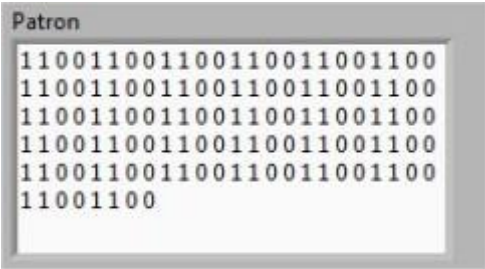


Figura 7.20: Patrón 0011 de 128 bits.



### 7.4.3. Demodulación de otros patrones

La demodulación de patrones con una frecuencia de cambio de símbolo mayor que la de los dos patrones anteriores, como el patrón 0101, no ha sido realizada correctamente en este TFG. En busca de la solución, se ha realizado la demodulación aumentando la tasa de muestreo del sistema, pero el resultado tampoco ha sido el esperado por lo que el problema debe radicar en la implementación del módulo *Demodulador.vi*.

Lamentablemente, la solución a este problema no ha podido ser implementada en este trabajo.

### 7.5. Respuesta del transmisor

Para analizar la respuesta del transmisor, se ha hecho uso del analizador de señales vectoriales y el software VSA instalado en el PC del laboratorio. Este analizador recibe constantemente la señal emitida por el USRP transmisor.

Como se indica en el apartado 6.6 de este trabajo, el transmisor implementado en este TFG emite de manera constante un slot FCCH, mientras que el receptor no haya detectado el patrón deseado. La figura 7.21 muestra una captura de pantalla realizada en el software VSA en la que se muestra la señal enviada por el USRP en el tiempo. En este caso el receptor aún no ha detectado ningún patrón por lo que solo se transmite el slot FCCH.



Figura 7.21: Señal recibida por el analizador vectorial. Un solo slot transmitido.

Se puede observar que aunque el periodo de tiempo que recoge el gráfico es levemente superior a la duración de una trama, solo se ve un slot, muestra de que solo se está transmitiendo el FCCH.

Una vez que el receptor detecta el patrón deseado, se activa la transmisión de un TCH en el slot TN3. La figura 7.22 muestra como el gráfico de la señal en tiempo posee ahora dos intervalos de tiempo transmitidos, siendo ambos los canales lógicos FCCH y TCH emitidos por el USRP.

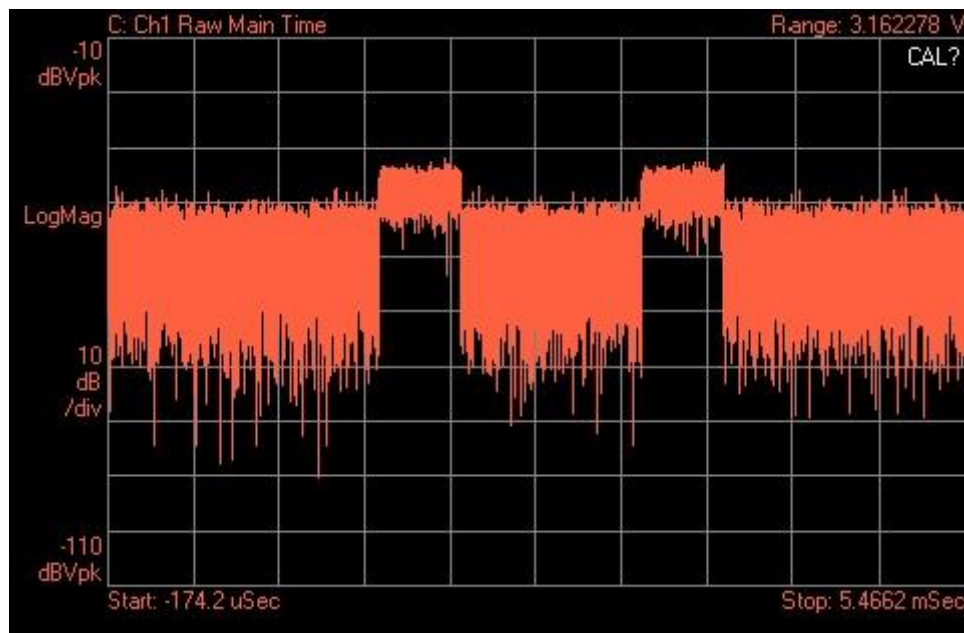


Figura 7.22: Señal recibida por el analizador vectorial. Dos slots transmitidos.

## 8. Presupuesto

En este apartado se detalla el presupuesto del desarrollo de este trabajo. Los costes tenidos en cuenta son los de personal y los del material empleado.

### 8.1. Costes material

Para realizar el cálculo de los costes de material se han tenido en cuenta tres variables:

- A: Tiempo de uso de los equipos
- B: Periodo de depreciación.
- C: Importe del material.

La fórmula para calcular la amortización de los equipos es la siguiente:

$$Coste = \frac{A}{B} * C$$

En la siguiente tabla se desglosan todos los costes de materiales.

	Importe (€)	Tiempo de uso (meses)	Periodo de depreciación (meses)	Coste (€)
<b>PC Portátil</b>	800	12	60	160
<b>PC Laboratorio</b>	600	1	60	120
<b>2 x NI USRP-2920 (Incluyendo cable MIMO y antenas)</b>	4990	6	60	499
<b>LabVIEW</b>	2920	12	60	584
<b>Agilent ESG E4438C</b>	86300	6	180	2876,66
<b>Agilent VSA 89600S</b>	165800	1	180	921,11
				<b>5160,77 €</b>

Tabla 9.1: Costes de material

El coste de material asciende a 5160,77€.

### 8.2. Costes de personal

Este trabajo ha sido realizado por un Graduado, con la colaboración de un Doctor Ingeniero y un Graduado. El número de horas imputables por el realizador del proyecto se estima con el siguiente cálculo:

$$22 \text{ días/mes} * 12 \text{ meses} * 1,5 \text{ horas/día} = 396 \text{ horas}$$

Estimando en una hora y media de trabajo al día durante un año.

El Graduado colaborador ha dedicado aproximadamente un 15% del tiempo, por lo que se le imputan 80 horas. El Doctor Ingeniero ha dedicado un 5% del tiempo haciendo un total de 20 horas. En la tabla 9.2 se recogen los costes de personal.

	Tiempo (horas)	Precio/hora estimado	Importe (€)
<b>Doctor Ingeniero</b>	20	100	2000
<b>Graduado cotutor</b>	60	30	1800
<b>Graduado</b>	396	30	11880
			<b>15680€</b>

Tabla 9.2: Costes de personal

### 8.3. Costes totales

Sumando los costes de personal a los costes de material, el coste total de este proyecto asciende al valor de: **20.840,77 €**.

## **9. Conclusiones**

### **9.1. Conclusiones generales**

Los resultados mostrados en el capítulo 7 muestran que se ha cumplido el objetivo de realizar un receptor estable GSM. Se ha logrado solucionar el grave problema de la pérdida de sincronización en tiempo logrando detectar la trama a recibir en todo momento de la ejecución.

También se ha plantado cara a las limitaciones de los transceptores USRP, realizando una adquisición controlada de muestras.

El objetivo de detectar la petición de comunicación de una estación móvil no se ha cumplido totalmente, si bien si ha sido hecho parcialmente. Pese a que el receptor implementado no está capacitado para detectar un canal RACH y contestar con un AGCH, si es capaz de detectar ciertos patrones de bits enviados por el móvil y contestar a la estación móvil con un TCH.

Por tanto, el programa implementado es capaz de mantener una comunicación con el móvil, pero esta comunicación no sigue el estándar GSM.

Por otra parte se ha cumplido el objetivo de realizar la implementación de la plataforma SDR usando exclusivamente el entorno de desarrollo LabVIEW, reduciendo el tiempo de procesamiento con ello.

### **9.2. Futuras líneas de trabajo**

Este trabajo marca una clara línea de trabajo. El siguiente paso a realizar debería ser la finalización de todos los objetivos marcados en este TFG, destacando la detección de una solicitud de acceso siguiendo el estándar GSM. Posteriormente se debería realizar una implementación extendida de la comunicación entre estación base y móvil, añadiendo más tipos de canales.

## 10. Referencias bibliográficas

- [1] GENOVÉS, BORJA. IMPLEMENTACION DE LA SEÑALIZACIÓN DE UNA ESTACIÓN BASE GSM. Tech. rep., Universidad Carlos III de Madrid, Departamento de Teoría de la Señal y Comunicaciones, 2013.
- [2] Nutaq Inc. A short history of software-defined radio (SDR) technology. [Fecha de consulta: 20 septiembre 2014]. Disponible en <http://nutaq.com/en/blog/short-history-software-defined-radio-sdr-technology>
- [3] GNURADIO. Página principal de GNU Radio. [Fecha de consulta: 20 septiembre 2014]. Disponible en <http://gnuradio.org/>
- [4] GNURADIO. Hardware - GNU Radio. [Fecha de consulta: 20 septiembre 2014]. Disponible en <http://gnuradio.org/redmine/projects/gnuradio/wiki/Hardware>
- [5] Wikipedia. Mercury Communications. [Fecha de consulta: 20 septiembre 2014]. Disponible en [http://en.wikipedia.org/wiki/Mercury\\_Communications](http://en.wikipedia.org/wiki/Mercury_Communications)
- [6] Medios Tecnologicos. [Imagen]. Disponible en <http://1.bp.blogspot.com/-LA7Tgs5X1Jk/UyNfvMzV0CI/AAAAAAAAAGk/fBam7GvpHbE/s1600/celdas-celulares.gif>
- [7] Kioskea. Estandar GSM. [Imagen]. Disponible en <http://static.commentcamarche.net/es.kioskea.net/pictures/telephonie-mobile-images-reseau-cellulaire.png>
- [8] Tutorialspoint. GSM - The mobile station. [Imagen]. Disponible en <http://www.tutorialspoint.com/images/gsm-mobile-station.gif>
- [9] Radio-Electronics.com. GSM Slot and Burst [Imagen]. Disponible en [http://www.radio-electronics.com/info/cellular telecomms/gsm\\_technical/gsm-radio-air-interface-slot-burst.php](http://www.radio-electronics.com/info/cellular telecomms/gsm_technical/gsm-radio-air-interface-slot-burst.php)
- [10] RUANO, ALVARO. DESPLIEGUE DE UN SISTEMA DE TELEFONÍA MOVIL GSM EN LAS COMARCAS DE TARRAGONA. Tech. rep., Universitat Rovira i Virgili, 2007.
- [11] National Instruments Inc. NI USRP-2920. [Fecha de consulta: 15 septiembre 2014]. Disponible en <http://sine.ni.com/nips/cds/view/p/lang/es/nid/209948>

## 11. Anexos

### Anexo 1: Cómo adquirir muestras con el USRP

El primer paso para poner en funcionamiento el receptor es iniciar la sesión de recepción. Para ello se hace uso del VI *niUSRP Open Rx Session.vi* proporcionado por las librerías instaladas con el driver del USRP. Como entrada del VI se indica la dirección IP del USRP y la función nos devuelve dos salidas, la sesión y el error. El error es un contenedor de tres elementos: el estado del error, su código y la fuente de este. La figura muestra el VI en cuestión:

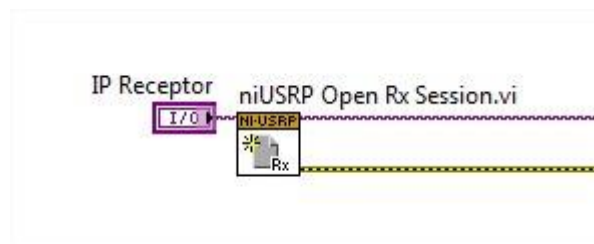


Figura 9.1: Inicio sesión USRP.

El USRP necesita que le indiquemos ciertos datos sobre la señal que debe recibir, por tanto el siguiente paso es configurar la señal a recibir. El VI *niUSRP Configure Signal.vi* se encarga de realizar este paso. Las entradas de este VI son:

- *Session handle*: la sesión ya iniciada del USRP, es decir, la salida del vi que inicia la sesión.
- *I/Q rate*: el ratio de los datos I/Q en banda base especificados en muestras por segundo (S/s).
- *Carrier frequency*: la frecuencia portadora especificada en Hz.
- *Gain*: la ganancia a aplicar a la señal recibida, expresada en dB.
- *Active antenna*: la antena del USRP que se desea usar para la recepción. Puede ser Rx1 o Rx2, dado que el USRP posee dos entradas para antenas.

La función tiene como salidas la sesión del instrumento y el error. Además también se proporcionan el IQ rate, la ganancia y la frecuencia reales con las que trabaja el USRP. De estas últimas tan solo usamos el IQ rate real, que en la práctica difiere un poco del establecido teóricamente.

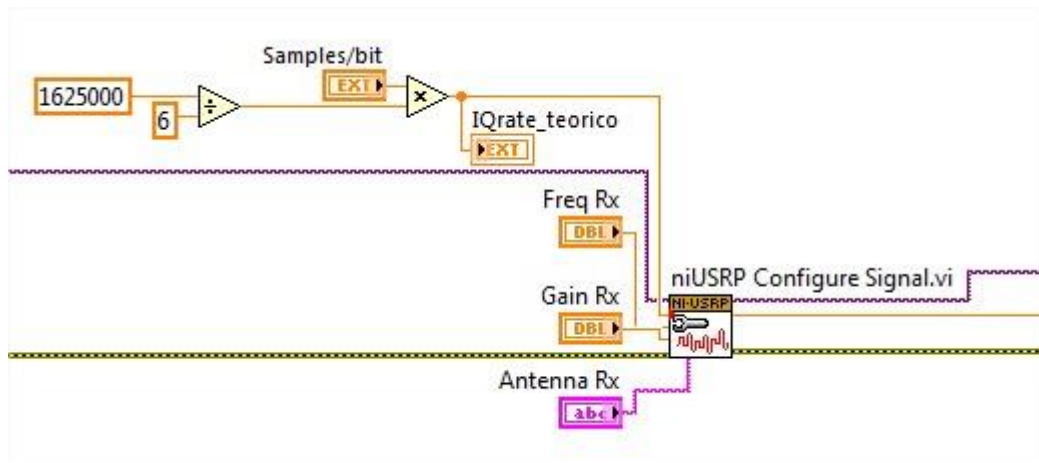


Figura 9.2: Diagrama de bloques configuración señal.

Después de configurar la señal, debemos establecer el número de muestras a adquirir. Por limitaciones del sistema no se puede establecer un número ilimitado de muestras. En consecuencia, se ha decidido establecer un control del flujo de datos recibidos con el fin de que el buffer no alcance el máximo de capacidad. El patrón usado ha sido adquirir 1000 tramas y una vez que se han procesado todas ellas se vuelven a adquirir otras 1000 tramas. Este método se explica con detalle en el apartado 6.3 de este trabajo. Por lo tanto el número de muestras a adquirir viene dado por el resultado de multiplicar las 1000 tramas por el tamaño en bits de cada trama y por el número de muestras por bit. El resultado es el siguiente:

$$1000\text{tramas} * 156,25b * 8\text{slots} * 4\text{Sps} = 5\,000\,000\,S$$

Para establecer si la adquisición de datos es finita o no y el número de muestras a tomar, debemos hacer uso del vi *niUSRP Configure Number of Samples.vi*. Las entradas de esta función son la sesión y el error dados por el vi anterior, una entrada booleana que indica si la adquisición será finita y el número de muestras a tomar.

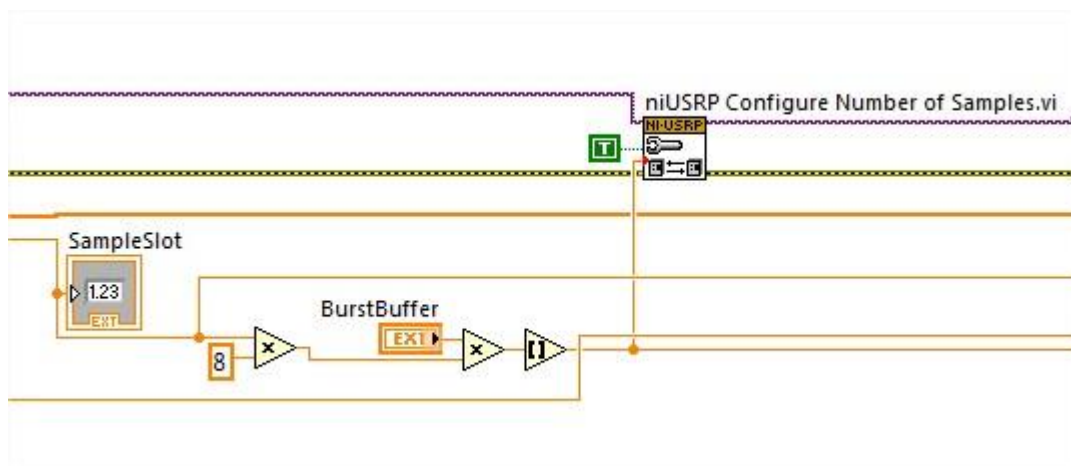


Figura 9.3: Diagrama de bloques configuración número muestras.



Tras este paso el receptor ya está preparado para iniciar la recepción de datos. El vi *niUSRP Initiate.vi* se encarga de ejecutar el receptor con la configuración preestablecida.



Figura 9.4: Esquema función *Initiate.vi*.

Finalmente, para leer los datos guardados en el buffer hay que hacer uso de la función *niUSRP Fetch Rx Data (poly).vi*. Este vi recibe como entradas la sesión y error, tal como los vi anteriores, el número de muestras a leer del buffer y un timeout que indica el tiempo de espera antes de devolver un error en caso de que no se consiga realizar la lectura. Las salidas son la sesión y error, los datos y la marca de tiempo de la primera muestra.

Las muestras son devueltas en un *cluster* que contiene un vector con los datos adquiridos, un número que indica el salto de tiempo entre muestras consecutivas y la marca de tiempo de la adquisición.

En este trabajo el número de muestras a leer del buffer ha sido comúnmente el número de muestras que contiene una trama entera, es decir ocho slots de tiempo.

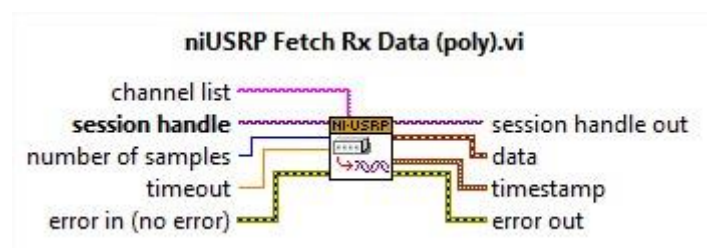


Figura 9.5: Esquema función *Fetch Rx Data (poly).vi*.

Para leer todos los datos adquiridos hay que ejecutar el vi anterior tantas veces como se desee hasta que no queden muestras en el buffer. Una vez llegados a ese punto hay dos opciones, dependiendo si se desea realizar una nueva recepción de datos o finalizar la comunicación.

En caso de que se quiera volver a adquirir datos, se deben leer todas las muestras adquiridas en el buffer o ejecutar el vi *niUSRP Abort.vi*, que aborta la adquisición de datos previa. Posteriormente hay que volver a ejecutar el vi *niUSRP Initiate.vi*.

Para finalizar la recepción, LabVIEW proporciona el vi *niUSRP Close Session.vi* que cierra la sesión iniciada en el USRP y devuelve los errores surgidos durante la sesión.